EXTENDED KALMAN FILTERING APPLIED TO THE POSITION LOCATING AND REPORTING SYSTEM (PLRS)

Bernard M. de Mahy

N Y KNOX LIBRARY
MON POSTGRADUATE SCHOOL
MONTEREY, CALIF. 93940

NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

EXTENDED KALMAN FILTERING APPLIED TO THE POSITION LOCATING AND REPORTING SYSTEM (PLRS)

by

Bernard M. de Mahy, Jr.

December 1976

Thesis Advisor: H. A. Titus

Approved for public release; distribution unlimited.

T177131

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle) Extended Kalman Filtering Applied to the Position Locating and Reporting System (PLRS)	5. TYPE OF REPORT & PERIOD COVERED Master's Thesis December 1976
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Bernard M. de Mahy, Jr.	8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940	12. REPORT DATE December 1976
	13. NUMBER OF PAGES 59
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) Naval Postgraduate School Monterey, California 93940	15. SECURITY CLASS. (of this report) Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

PLRS
Position Locating Reporting System
Kalman Filter

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The Marine Corps and Army are developing a Position Locating Reporting System to aid the battlefield commander in locating his assets during battle.

This study has applied Extended Kalman Filtering techniques to that problem, evolving from a simple Extended Kalman Filter Observer to three moving observers, whose position is uncertain, estimating the position of another unit.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
(Page 1) S/N 0102-014-6601

EXTENDED KALMAN FILTERING APPLIED TO THE POSITION LOCATING AND REPORTING SYSTEM (PLRS)

by

Bernard M. de Mahy, Jr.
Captain, United States Marine Corps
B.S., University of Southwestern Louisiana, 1969

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the
NAVAL POSTGRADUATE SCHOOL
December 1976

Thesis
D29845
C.1

DUDLEY KNOX LIBRARY
NAVAL POSTGRADUATE SCHOOL

ABSTRACT

The Marine Corps and Army are developing a Position Locating Reporting System to aid the battlefield commander in locating his assets during battle.

This study has applied Extended Kalman Filtering techniques to that problem, evolving from a simple Extended Kalman Filter Observer to three moving observers, whose position is uncertain, estimating the position of another unit.

TABLE OF CONTENTS

LIST OF FIGURES.................................... 6

LIST OF TABLES..................................... 7

ACKNOWLEDGEMENTS................................... 8

I. INTRODUCTION.................................... 9

II. EXTENDED KALMAN FILTERING...................... 12

III. CHOOSING THE BEST RANGER...................... 17

IV. PLRS SIMULATION................................ 19

A. TWO RANGING UNITS............................... 19

B. THREE RANGING UNITS............................. 23

C. RANGING UNITS IN MOTION......................... 26

D. SOURCE OF MEASUREMENT NOISE..................... 30

V. CONCLUSIONS..................................... 35

COMPUTER PROGRAM................................... 36

LIST OF REFERENCES................................. 58

INITIAL DISTRIBUTION LIST.......................... 59

LIST OF FIGURES

1. Consecutive Updates Will Reduce the Degree of Uncertainity in the Direction of the Ranging Unit 11

2. PLRS Simulation - A Jet in a Constant 10Km Radius Turn Flying Between Two Stationary Rangers 21

3. PLRS Simulation - A Jet in a Constant 10Km Radius Turn Flying Among Three Stationary Rangers 24

4. PLRS Simulation - A Jet in a Constant 10Km Radius Turn Flying Among Three Moving Rangers 28

5. The Geometry of the Covariance of Measurement Noise Calculations 32

6. PLRS Simulation - A Jet in a Constant 10Km Radius Turn Flying Among Three Moving Rangers With Position Uncertainty 33

LIST OF TABLES

1. The Ranger Chosen at Each Time for the Two Stationary Ranger Simulation 22

2. The Ranger Chosen at Each Time for the Three Stationary Ranger Simulation 25

3. The Ranger Chosen at Each Time for the Three Moving Ranger Simulation 29

4. The Ranger Chosen at Each Time for the Three Moving Rangers With Position Uncertainty Simulation 34

ACKNOWLEDGEMENTS

The author wishes to acknowledge the assistance of Professors H. A. Titus and D. E. Kirk.

The author, moreover, wishes to acknowledge the love and sacrafices that he recieved from his wife, Pam, and children, Marc and Jennifer, while he conducted this study and hereby dedicates this Thesis to them.

I. INTRODUCTION

The precise location of all assets in and about the battle area is of prime importance to the tactical Marine Commander. In the past locating has had to depend on the individual knowing his own position and being able to report it through radio links to higher command. This system suffered from the limitations of terrain, daylight, weather, and the volume of radio traffic during battle.

To alleviate these shortcomings the Marine Corps and Army are investigating a Position Locating and Reporting System (PLRS) to collect, process, and, display the location of units, vehicles, and aircraft in and about the battle area.

The PLRS consists of field units and a master unit. The field unit is compact enough to be carried in the field by a man, vehicle, or aircraft. These units will determine the range to other field units in the area and report this information to the master unit for processing and display. The range information is determined by measuring the time required to send a signal from one unit to another and back again plus some "system" delay. When a unit's position is being updated it is referred to as the " Update" unit; and all others are referred to as "Ranging" units.

In a previous study in this area,[1], tests were conducted to investigate the use of the error ellipse in visually displaying the degree of uncertainty of the position of an update unit and the effect of numerous updates on reducing that degree of uncertainty It was

found that the degree of uncertainty is reduced in the direction of the ranging unit with consecutive updates as shown in Fig 1 taken from that study.

That study also simulated one jet aircraft flying Mach 1 in a constant radius turn as an update unit being ranged on by two stationary ranging units to explore the proper random forcing excitation covariance necessary for adequate filter performance.

It is the intent of this study to further expand the simulation begun in the previous work by adding an additional ranging unit, allowing the movement of the ranging units, and considering the effect of ranging from a unit whose position is not known exactly

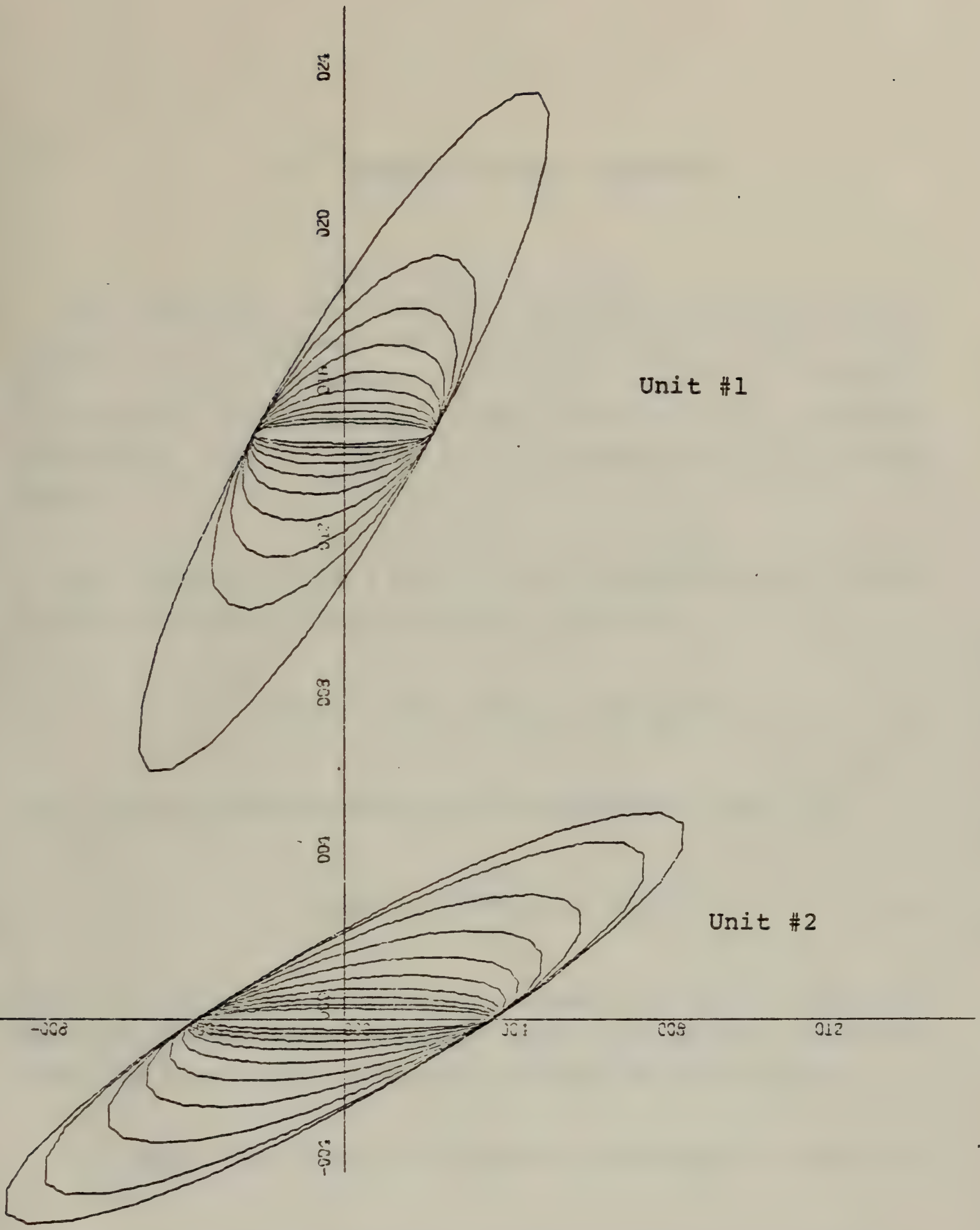

Figure 1 - CONSECUTIVE UPDATES WILL REDUCE THE DEGREE OF UNCERTAINTY IN THE DIRECTION OF THE RANGING UNIT

II. EXTENDED KALMAN FILTERING

The Extended Kalman Filter is widely documented and no attempt at a development of that theory will be made in this work. A brief treatment has been included to establish nomenclature and formulas used. For a more complete development one is referred to reference [2] or similar texts.

As defined in this work, PLRS is described by a set of discrete, linear, cartesian system equations

$$\underline{x}(k+1) = \underline{\phi}(k)\ \underline{x}(k) + \underline{\Gamma}(k)\ \underline{w}(k) \qquad (1)$$

and a set of discrete non-linear measurement equations

$$\underline{z}(k) = \underline{m}(x(k),k) + \underline{v}(k) \qquad (2)$$

where $\underline{\Phi}$ and $\underline{\Gamma}$ are linear functions and $\underline{m}$ is a nonlinear function of the state variables $\underline{x}(k)$; $\underline{w}(k)$ is the excitation noise and $\underline{v}(k)$ is the measurement noise of the system.

The plant noises are considered uncorrelated, zero-mean, and white.

The non-linear measurement equations can be linearized by expanding equation (2) around the best estimate at time k and using the first-order terms yielding

$$\underline{z}(k) = \underline{H}(k)\ \underline{x}(k) + \underline{v}(k)$$

where

$$H(k) = \frac{\partial \underline{m}}{\partial \underline{x}}\ \underline{x}=\hat{\underline{x}}(k/k-1) \tag{3}$$

$\tilde{\underline{x}}(k/k)$ is the estimated value of the state at k after the k^{th} measurement and $\hat{\underline{x}}(k/k-1)$ is the predicted value of the state at time k before the k^{th} measurement.

The state error vector is

$$\hat{\underline{x}}'(k/k) = \hat{\underline{x}}(k/k) - \hat{\underline{x}}(k)$$

and the predicted error vector is

$$\hat{\underline{x}}'(k/k-1) = \hat{\underline{x}}(k/k-1) - \underline{x}(k)$$

The covariance of the state error matrix is

$$P(k/k) = E[\hat{\underline{x}}'(k/k)\ \hat{\underline{x}}'^{T}(k/k)]$$

and the predicted covariance of the state error matrix is

$$P(k/k-1) = E[\tilde{\underline{x}}'(k/k-1)\ \hat{\underline{x}}'^{T}(k/k-1)]\ .$$

The state excitation matrix is

$$Q(k) = E[\underline{\Gamma}(k)\ \underline{w}(k)\ \underline{w}^{T}(k)\ \underline{\Gamma}^{T}(k)]$$

and the measurement noise covariance matrix is

$$R(k) = E[\underline{v}(k)\ \underline{v}^{T}(k)]\ .$$

The equations that made up the Kalman Filter used in this work are as follows:

$$P(k/k-1) = \underline{\phi}(k)\ P(k/k)\ \underline{\phi}^{T}(k) + Q(k)$$

$$G(k) = P(k/k-1)\ H^{T}(k)[H(k)\ P(k/k-1)\ H^{T}(k) + R(k)]^{-1}$$

$$P(k/k) = [I - G(k)\ H(k)]P(k/k-1)$$

$$\hat{\underline{x}}(k/k) = \hat{\underline{x}}(k/k-1) + G(k)[\underline{z}(k) - H(k)\ \underline{x}(k/k-1)]$$

$$\hat{\underline{x}}(k/k-1) = \underline{\phi}(k)\ \hat{\underline{x}}(k/k)$$

$$\underline{z}(k) = \underline{m}(\underline{x}(k/k-1),k)$$

Since the only observations in this system are ranges,

the observation equation is

$$\underline{z}(k) = [x^2(k) + y^2(k)]^{1/2} \quad ;$$

and from equation (3) we get

$$H(k) = \frac{x(k)}{x^2(k) + y^2(k)} \quad 0 \quad \frac{y(k)}{x^2(k) + y^2(k)} \quad 0 \quad .$$

The covariance of estimation error, P, is and expression of the uncertainity in the estimation of the states. Considering only the estimation's position error, $P_{position}$ can be expressed as

$$P_{position} = \begin{matrix} \sigma_x^2 & \sigma_x\sigma_y \\ \sigma_y\sigma_x & \sigma_y^2 \end{matrix}$$

Since the position estimation error is normally distributed, a curve of constant error probability can be defined by using the exponent of the normal distribution,

$$\frac{x^2}{\sigma_x^2} - \frac{2r_{xy}\,xy}{\sigma_x\sigma_y} + \frac{y^2}{\sigma_y^2}$$

This curve defines an ellipse. Graphically, for the given probability, the estimation may be anywhere in that ellipse.

III. CHOOSING THE BEST RANGER

To move from a simple Kalman Filter observer to the PLRS model, the first problem encountered was to choose the best ranger from which to take the measurement. In the previous work ,[1], it was shown that the most useful measurement, the one causing the most reduction in the error ellipse, is obtained by observing the update unit from a point aligned with the major axis of the error ellipse of the update unit.

To find the ranger most closely aligned with the major axis of the update unit's error ellipse the orientation of the error ellipse must first be found using the following equation.

$$\theta = \frac{1}{2} \tan^{-1} \frac{2 \operatorname{Cov}(x,y)}{\sigma_x^2 - \sigma_y^2}$$

This angle(θ) gives the angle between -90° and 90° that the x-axis of the ellipse makes with the x-axis of the co-ordinate system. Looking at the ellipse in this new posture one can find the new "Uncorrelated" variances that define the major and minor axes.

$$\sigma_x^{2'} = \frac{\sigma^2 + \sigma^2}{2} + \frac{\operatorname{Cov}(x,y)}{\operatorname{Sin} 2\theta} ,$$

$$\sigma_y^{2'} = \frac{\sigma^2 + \sigma^2}{2} - \frac{\operatorname{Cov}(x,y)}{\operatorname{Sin} 2\theta} ,$$

If σ_x^2 is greater than σ_y^2 the x-axis of the error ellipse is the major axis and θ is the angle we seek. If σ_y^2 is greater than σ_x^2 then the y-axis of the error ellipse is the major axis and the angle we seek is $\theta + 90°$.

The bearing of the update unit from the ranger must then be found and it is simply

$$\beta = \mathrm{Tan}^{-1} \frac{y_u - y_R}{x_u - x_R} .$$

The absolute value of the difference between θ, after proper correction, and β was chosen as the best alignment indicator; but to be aligned and to be 180° out of alignment is of equal value; therefore the absolute value of the cosine of the differences was used as the alignment indicator and the ranger found to have the largest indicator was chosen as the ranging unit for that measurement.

IV. PLRS SIMULATION

A. TWO RANGING UNITS

In previous work,[1], the PLRS simulation was setup for a jet aircraft flying Mach 1 in a constant 10 Km turn about the origin to act as the update unit for all measurements. Two stationary ranging units were placed at the origin and at 10Km north, 10Km east. Using a one second sample interval, the jet was described by the following matrices:

$$\phi = \begin{matrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{matrix}$$

$$\Gamma = \begin{matrix} 0.5 & 0 \\ 1 & 0 \\ 0 & 0.5 \\ 0 & 1 \end{matrix}$$

Its initial state was

$$\underline{x} = \begin{matrix} 0 \\ 0.333 \text{ Km/s} \\ 10 \text{ Km} \\ 0 \end{matrix}$$

Its initial covariance of error matrix, measurement noise covariance, and excitation forcing matrix were

$$P(1/0) = \begin{matrix} 10^{-4} & 0 & 10^{-4} & 0 \\ 0 & 10^{-4} & 0 & 0 \\ 10^{-4} & 0 & 10^{-4} & 0 \\ 0 & 0 & 0 & 10^{-4} \end{matrix}$$

and

$$R = 10^{-4}$$

with

$$Q = \begin{matrix} 2.5\times10^{-5} & 5\times10^{-5} & 0 & 0 \\ 5\times10^{-5} & 10^{-4} & 0 & 0 \\ 0 & 0 & 2.5\times10^{-5} & 5\times10^{-5} \\ 0 & 0 & 5\times10^{-5} & 10^{-4} \end{matrix}$$

Fig 2 is a display of its final runs. The filter tracked accurately and the error ellipses shown are twenty times their actual size to make them visible. Table 1 shows which was the ranging unit at each measurement time.

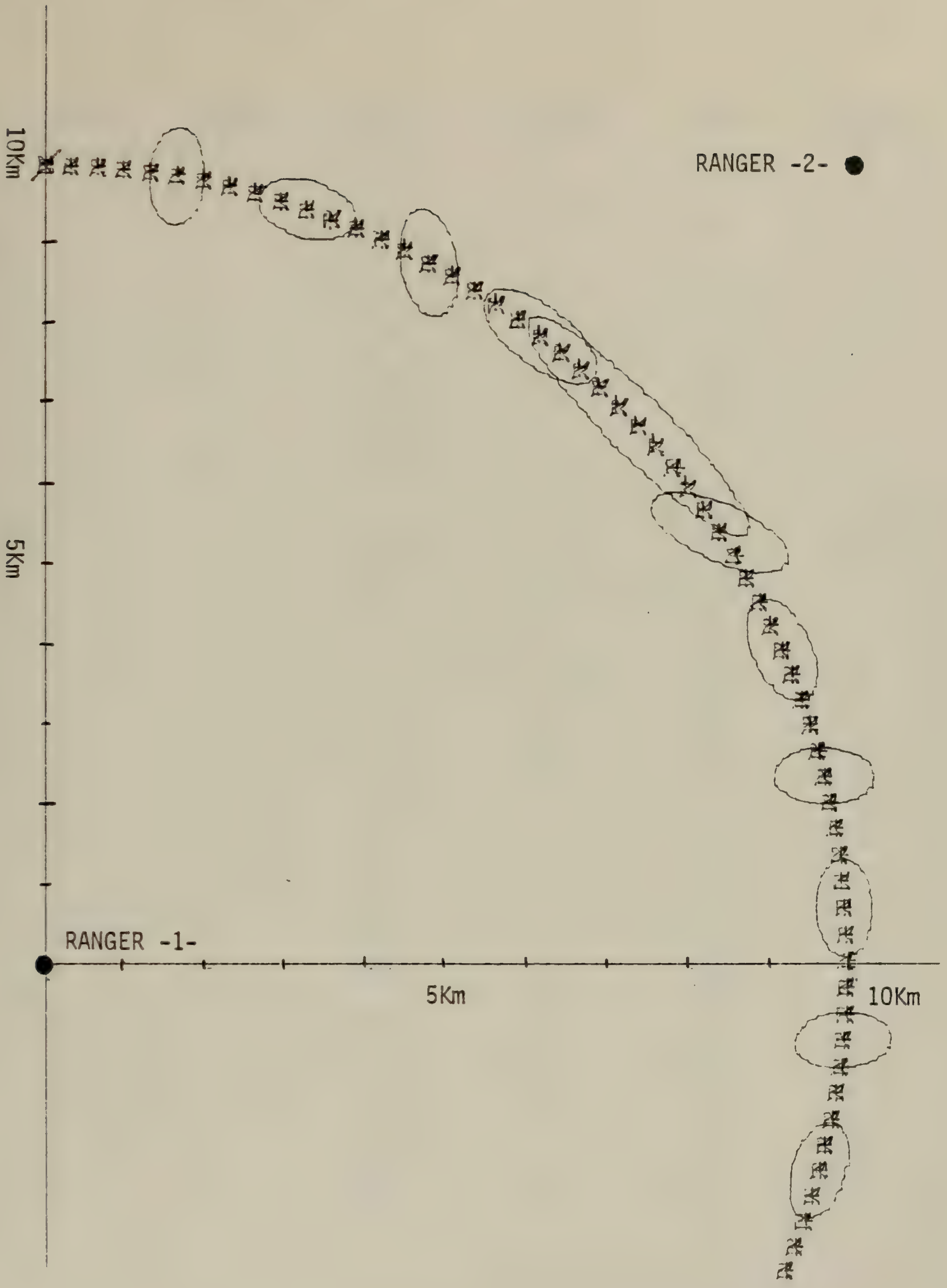

Figure 2 - PLRS SIMULATION - A JET IN A CONSTANT 10KM RADIUS TURN FLYING BETWEEN TWO STATIONARY RANGERS

TIME	RANGER	TIME	RANGER	TIME	RANGER
1	2	21	2	41	1
2	1	22	1	42	2
3	2	23	2	43	1
4	1	24	1	44	2
5	2	25	1	45	1
6	1	26	2	46	2
7	2	27	1	47	1
8	1	28	2	48	2
9	2	29	1	49	1
10	1	30	2	50	2
11	2	31	1	51	1
12	1	32	2	52	2
13	2	33	1	53	1
14	1	34	2	54	2
15	2	35	1	55	1
16	1	36	2	56	2
17	2	37	1	57	1
18	1	38	2	58	2
19	2	39	1	59	1
20	1	40	2	60	2

TABLE 1 - THE RANGER CHOSEN AT EACH TIME FOR THE PLRS TWO STATIONARY RANGER SIMULATION

B. THREE RANGING UNITS

The first step of this study was to add a third ranging unit at 0 north, 10Km east. The algorithm was enlarged to include the additional unit and its comparison with the alignment indicators of the other ranging units.

It can be seen in Fig 3 that the size of the error ellipses were reduced in size in the mid-range area where the jet and the two original units were in line; and the third ranger provides the triangular measurement.

Table 2 shows the ranging unit chosen for the measurement at each time k.

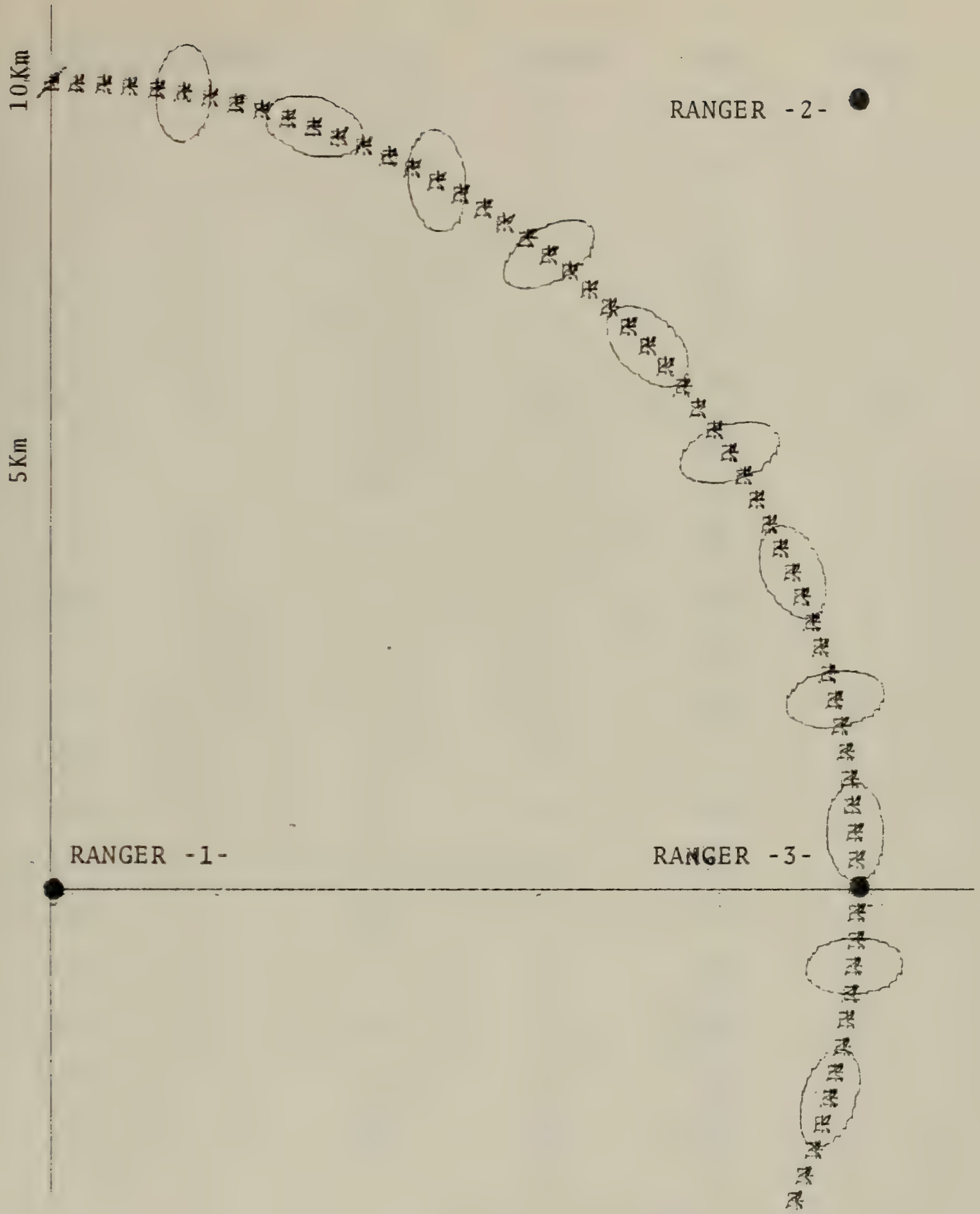

Figure 3 - PLRS SIMULATION - A JET IN A CONSTANT 10KM RADIUS TURN FLYING AMONG THREE STATIONARY RANGERS

TIME	RANGER	TIME	RANGER	TIME	RANGER
1	2	21	2	41	1
2	1	22	3	42	3
3	2	23	2	43	1
4	1	24	3	44	3
5	2	25	1	45	1
6	1	26	3	46	3
7	2	27	1	47	1
8	1	28	3	48	2
9	2	29	1	49	1
10	1	30	3	50	2
11	2	31	1	51	1
12	1	32	3	52	3
13	2	33	1	53	1
14	3	34	3	54	3
15	2	35	1	55	1
16	3	36	3	56	3
17	2	37	1	57	1
18	3	38	3	58	3
19	2	39	1	59	1
20	3	40	3	60	3

TABLE 2 - THE RANGER CHOSEN AT EACH TIME FOR THE THREE STATIONARY RANGER SIMULATION

C. RANGING UNITS IN MOTION

In the second step the rangers are given motion. The rangers at the origin and at 10Km north, 10Km east were to move north and south respectively at 3Kts as infantrymen. The ranger at 0 north, 10Km east was to move west at 120Kts as a helicopter. Again using one second sample intervals, their motion was defined using discrete linear state equations

$$x(k+1) = \phi(k)\ x(k) \ ,$$

where

$$\phi = \begin{matrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{matrix}$$

with the initial states shown below;

$$x = \begin{matrix} 0 \\ 0 \\ 0 \\ 1.67\text{x}10^3 \text{ Km/s} \end{matrix} \qquad \text{INFANTRYMAN}$$

$$x = \begin{matrix} 10 \\ 0 \\ 10 \\ -1.67\text{x}10^3 \text{ Km/s} \end{matrix} \qquad \text{INFANTRYMAN}$$

$$x = \begin{matrix} 12 \\ -5.555\text{x}10^{2}\ \text{Km/s} \\ 0 \\ 0 \end{matrix} \quad \text{HELICOPTER}$$

It can be seen in Fig 4 that no system depreciation resulted from the motion of the rangers, Table 3 shows the ranging unit chosen for the measurement at each time,

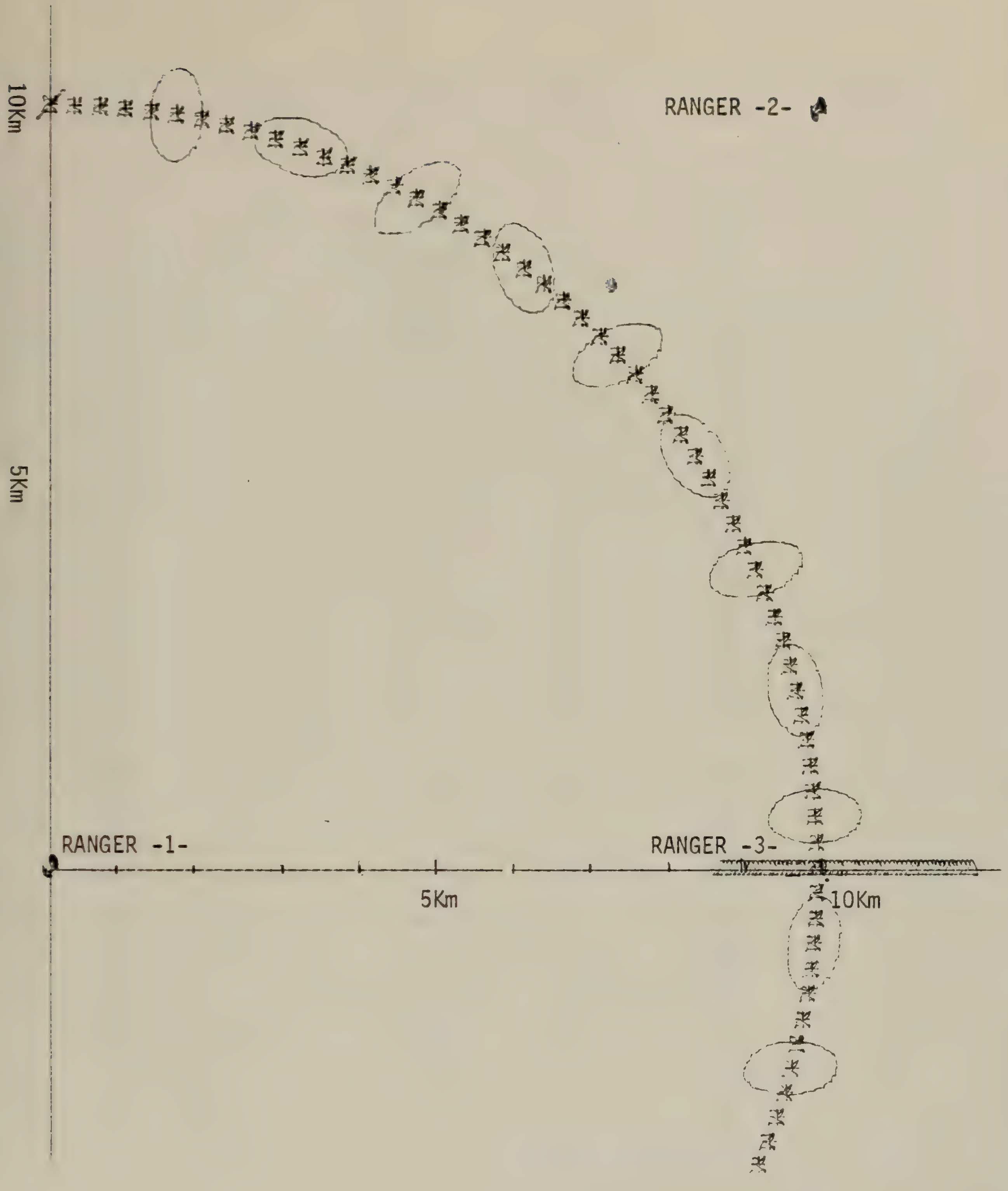

Figure 4 - PLRS SIMULATION - A JET IN A CONSTANT 10KM RADIUS TURN FLYING AMONG THREE MOVING RANGERS

TIME	RANGER	TIME	RANGER	TIME	RANGER
1	2	21	3	41	3
2	1	22	2	42	1
3	2	23	3	43	2
4	1	24	2	44	1
5	2	25	3	45	2
6	1	26	1	46	1
7	2	27	3	47	2
8	1	28	1	48	1
9	2	29	3	49	2
10	1	30	1	50	1
11	2	31	3	51	2
12	1	32	1	52	1
13	3	33	3	53	2
14	1	34	1	54	1
15	3	35	3	55	2
16	2	36	1	56	1
17	3	37	3	57	2
18	2	38	1	58	1
19	3	39	3	59	2
20	2	40	1	60	1

TABLE 3 - THE RANGER CHOSEN AT EACH TIME FOR THE THREE MOVING RANGER SIMULATION

D. SOURCE OF MEASUREMENT NOISE

In the above simulations the position of the ranging unit has been assumed to be exact; while in actual application the ranging units will have covariances of estimation error defining an error ellipse; and the ranging unit might be anywhere within that ellipse. To bring this position uncertainty into the simulation, the radius of the error ellipse along the bearing from the ranging unit to the update unit was defined as the covariance of measurement error.

The equation for the radius of an ellipse is a function of the major axis, the minor axis, and the angle at which the measurement is made. To find the measurement noise covariance, or the ellipse radius, $\sigma_x^{2'}$ and $\sigma_y^{2'}$ must be compared and the larger defined as Mj, the major axis, and the smaller defined as Mn, the minor axis. The angle, α , at which the radius is determined is measured from the major axis and thus is calculated as the difference between θ and β. Fig 5 shows the geometry of the calculation of the covariance of measurement noise. The equation for R and the radius squared of the ellipse is:

$$R = r^2 = \frac{Mj\ Mn}{Mj\ \mathrm{Sin}^2\alpha + Mn\ \mathrm{Cos}^2\alpha}$$

It can be seen in Fig 6 that performance was improved slightly using the covariance of estimation error as the sole source of measurement noise.

Table 4 shows the ranger chosen at each time for the three moving rangers with position uncertaininity simulation.

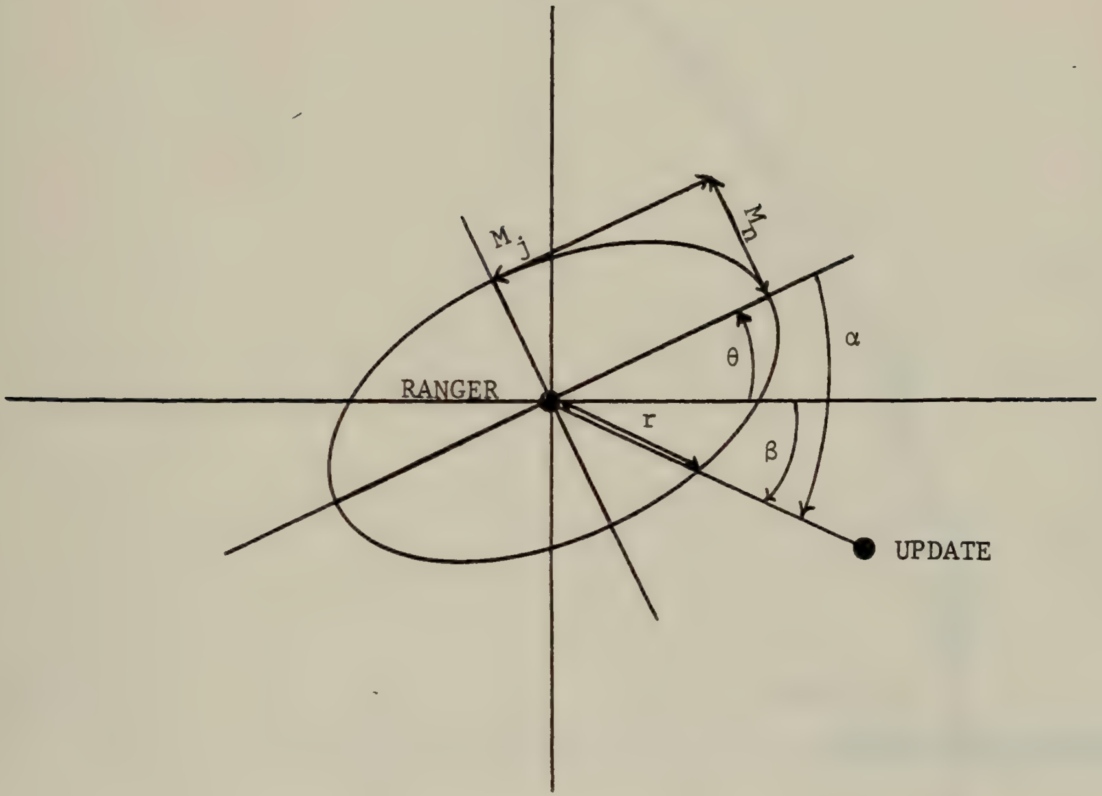

Figure 5 - r IS THE RADIUS OF THE ERROR ELLIPSE - $r^2 = R$ IS THE COVARIANCE OF MEASUREMENT NOISE

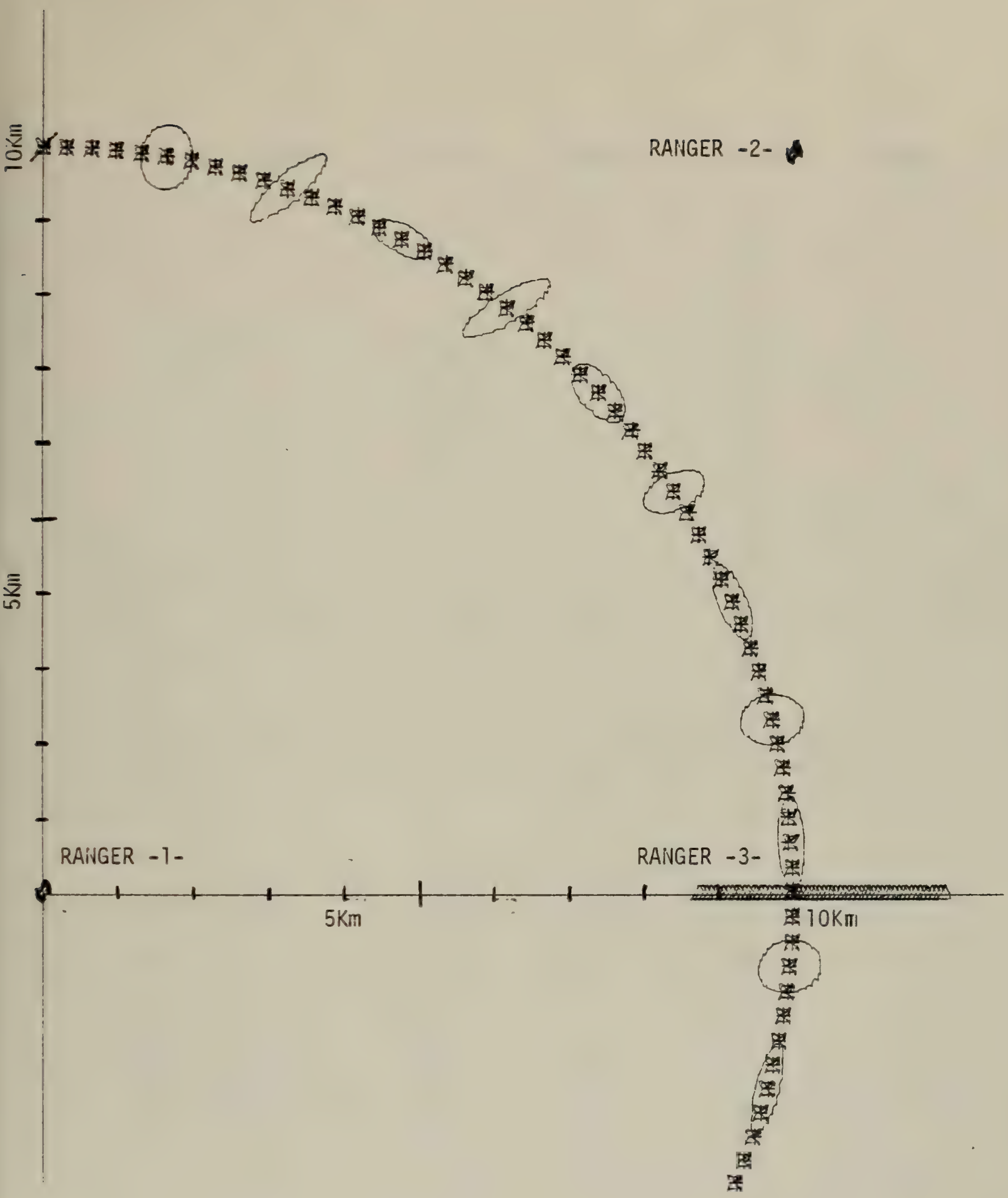

Figure 6 - PLRS SIMULATION - A JET IN A CONSTANT 10KM RADIUS TURN FLYING AMONG THREE MOVING RANGERS WITH POSITION UNCERTAINTY

TIME	RANGER	TIME	RANGER	TIME	RANGER
1	2	21	2	41	1
2	1	22	3	42	2
3	2	23	2	43	1
4	1	24	3	44	2
5	2	25	1	45	1
6	1	26	3	46	2
7	2	27	1	47	1
8	1	28	3	48	2
9	2	29	1	49	1
10	3	30	3	50	2
11	1	31	1	51	1
12	3	32	3	52	2
13	1	33	1	53	1
14	3	34	3	54	2
15	1	35	1	55	1
16	3	36	3	56	2
17	1	37	1	57	1
18	3	38	3	58	2
19	2	39	1	59	1
20	3	40	3	60	2

TABLE 4 - THE RANGER CHOSEN AT EACH TIME FOR THE THREE MOVING RANGERS WITH POSITION UNCERTAINTY SIMULATION

V. CONCLUSION

The placement of the third ranger showed the value of triangulation of the rangers. The closer to normal the bearings of the rangers are to each other, the better the results of consecutive ranges.

The allowance for motion and the representation of the ranger's position uncertainity as the source of measurement error were important steps toward full simulation of the system; and they were accomplished without degradation of performance.

A better simulation may be to represent the measurement error as the ranger's position uncertainty plus some system measurement error.

Still to be accomplished is the ability to update all units at each ranging, and to provide a gating system that will demand more frequent updates for faster moving units and less frequent updates for slower units.

A program listing of the three moving rangers with position uncertainity is included with an annotated data deck.

```
// EXEC FORTCLGP,REGION.GO=200K
//FORT.SYSIN DD *
C         PLRS
C                                                                       MCSP0006
      REAL*8 GAMMA,COVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,G,PKK,Q,EI,PR,PRRCH3*****
       CCMMON EI(4,4),Q(4,4),G(4,4),PKK(4,4),GAMMA(4,4),COVW(4,4),      MCSP0008
     1TEMP(4,4),TEMP1(4,4),TEMP2(4,4),H(4,4),PKKM1(4,4),R(4,4),PHI(4,4),MCSP0009
     2VAR(4,4,60),GKS(4,4,60),PKKS(4,4,60),XM(4,4,60),ERR(4,4,60),      CH200002
     3GAMMAS(4,4),PHIS(4,4),XS(4,4,60),HS(4,4),GK(4,4),SIGW(4),X(4),    CH200003
     4SIGXZ(4),XZMEAN(4),XHKK(4),XHKKM1(4),VTMP(4),Z(4),V(4),SIGV(4),   MCSP0012
     5XHATZ(4,4),XZ(60),YZ(60),PX(10),PY(10),PR(4,4,4),                 CH200004
     6N,NSAM,IQ,M,ITER,ITRK,IN,ISTAT,K,ITRO,IXZ,IV,IW,IEST,ND,NR        CH2
       DIMENSION XP(80),YP(80),PRR(4,4)                                 CH3*****
C                                                                       MCSP0016
C                                                                       MCSP0017
C     N=ORDER OF SYSTEM MODEL AND FILTER (DIMENSICN OF X,XHAT)          MCSP0018
C                                                                       MCSP0019
C     M=NUMBER OF MEASUREMENTS (DIMENSION OF THE VECTOR Z)              MCSP0020
C                                                                       MCSP0021
C     IN=NUMBER OF INPUT RANDOM FORCING FCNS (=DIMENSION OF W)          MCSP0022
C                                                                       MCSP0023
C     NSAM=NUMBER OF TIME SAMPLES                                       MCSP0024
C                                                                       MCSP0025
C     NENS=NUMBER OF MEMBERS IN ENSEMBLE                                MCSP0026
C                                                                       MCSP0027
      READ (5,81) N,M,IN,NSAM,NENS,NR                                   CH200005
C                                                                       MCSP0029
C                                                                       MCSP0030
      READ (5,82) ND                                                    MCSP0031
C     THE VALUE OF ND READ IN MUST EQUAL THE ROW (AND COLUMN) DIMENSION MCSP0032
C     SPECIFIED FOR THE SQUARE MATRIX "TEMP1", E.G. IF TEMP1(3,3) IS    MCSP0033
C     SPECIFIED IN THE COMMON STATEMENT "ND" MUST BE EQUAL TO 3.        MCSP0034
C                                                                       MCSP0068
C     IPRT=0 -- SOME OR ALL OUTPUT DATA IS PRINTED                      MCSP0069
C                                                                       MCSP0070
C     IPLT=0 -- SOME OR ALL OUTPUT DATA IS PLOTTED                      MCSP0071
C                                                                       MCSP0073
      READ (5,84) IPRT,IPLT                                             MCSP0072
C                                                                       MCSP0035
      CALL OVFLOW                                                       MCSP0108
      IW = 6395217                                                      MCSP0109
      IV = 1936748                                                      MCSP0110
      IXZ = 135769                                                      MCSP0111
C                                                                       MCSP0112
C                                                                       MCSP0199
C     THE FOLLOWING SECTION READS THE SPECIFIED INPUT MATRICES          MCSP0200
C                                                                       MCSP0201
      CALL MREAD (PHI,N,N)                                              MCSP0203
```

```
C                                                                       MCSP0204
      DC 23 I=1,N                                                       MCSP0205
      DO 23 J=1,N                                                       MCSP0207
   23 PHIS(I,J) = PHI(I,J)                                              MCSP0208
      WRITE (6,131)                                                     MCSP0210
      CALL MWRITE (PHI,N,N)                                             MCSP0211
C                                                                       MCSP0212
C                                                                       MCSP0213
      CALL MREAD (H,M,N)                                                MCSP0215
      DO 25 I=1,M                                                       MCSP0217
      DO 25 J=1,N                                                       MCSP0219
   25 HS(I,J) = H(I,J)                                                  MCSP0220
      WRITE (6,132)                                                     MCSP0222
      CALL MWRITE (H,M,N)                                               MCSP0223
C                                                                       MCSP0224
C                                                                       MCSP0225
      CALL MREAD (R,M,M)                                                MCSP0227
      WRITE (6,133)                                                     MCSP0228
      CALL MWRITE (R,M,M)                                               MCSP0229
C                                                                       MCSP0230
C                                                                       MCSP0231
      CALL MREAD (COVW,IN,IN)                                           MCSP0233
      WRITE (6,134)                                                     MCSP0234
      CALL MWRITE (COVW,IN,IN)                                          MCSP0235
      CALL MREAD (GAMMA,N,IN)                                           MCSP0244
C                                                                       MCSP0245
      DO 30 I=1,N                                                       MCSP0246
      DO 30 J=1,IN                                                      MCSP0248
   30 GAMMAS(I,J) = GAMMA(I,J)                                          MCSP0249
      WRITE (6,136)                                                     MCSP0251
      CALL MWRITE (GAMMA,N,IN)                                          MCSP0252
C                                                                       MCSP0253
C                                                                       MCSP0254
      CALL MREAD (PKKM1,N,N)                                            MCSP0256
      WRITE (6,137)                                                     MCSP0257
      CALL MWRITE (PKKM1,N,N)                                           MCSP0258
C                                                                       MCSP0259
C
      DO 311 K=2,NR                                                     CH3****1
      CALL MREAD (PRR,N,N)                                              CH3****2
      DO 310 I=1,N                                                      CH3****3
      DO 310 J=1,N                                                      CH3****4
  310 PR(I,J,K) = PRR(I,J)                                              CH3****5
  311 CONTINUE                                                          CH3****6
C
C                                                                       MCSP0260
      CALL VREAD (SIGV,M)                                               MCSP0262
      WRITE (6,138)                                                     MCSP0263
```

```
      CALL VWRITE (SIGV,M)                                              MCSP0264
C                                                                       MCSP0265
C                                                                       MCSP0272
      DO 340 I=1,NR                                                     CH200006
      READ (5,144) (XHATZ(I,J),J=1,N)                                   CH200007
      WRITE (6,140)                                                     MCSP0275
  340 WRITE (6,146) (XHATZ(I,J),J=1,N)                                  CH200008
C                                                                       MCSP0277
C                                                                       MCSP0292
   36 DO 360 I=1,NR                                                     CH200009
      READ (5,144) (XS(I,J,1),J=1,N)                                    CH200010
C     INITIAL CONDITION HAS BEEN READ                                   MCSP0294
      WRITE (6,143)                                                     MCSP0295
  360 WRITE (6,146)(XS(I,J,1),J=1,N)                                    CH200011
C                                                                       MCSP0302
C                                                                       MCSP0247
   38 CALL TRACK                                                        MCSP0304
C                                                                       MCSP0250
C                                                                       MCSP0307
      DO 390 K=1,NR                                                     CH200012
   39 WRITE (6,145)                                                     MCSP0308
      WRITE (6,146) (XS(K,I,1),I=1,N)                                   CH200013
      WRITE (6,146) (XS(K,I,NSAM),I=1,N)                                CH200014
  390 CONTINUE                                                          CH200015
C
C                                                                       MCSP0313
C     THE FOLLOWING SECTION PREPARES FOR THE MONTE CARLO LOOP           MCSP0314
C                                                                       MCSP0221
C     FORM NXN IDENTITY MATRIX IN DOUBLE PRECISION                      MCSP0316
C                                                                       MCSP0317
      DO 41 I=1,N                                                       MCSP0318
      DO 41 J=1,N                                                       MCSP0320
      EI(I,J) = 0.D0                                                    MCSP0321
   41 IF (I.EQ.J) EI(I,J)=1.D0                                          MCSP0322
C                                                                       MCSP0323
C     GIVEN THE MATRIX GAMMA AND THE COVARIANCE OF W COMPUTE Q          MCSP0324
C     USING DOUBLE PRECISION ARITHMETIC                                 MCSP0325
C                                                                       MCSP0319
      CALL QMAT                                                         MCSP0327
      WRITE (6,135)                                                     MCSP0328
      CALL MWRITE (Q,N,N)                                               MCSP0329
C                                                                       MCSP0356
C     SET UP ARRAYS FOR COMPUTING STATISTICS                            MCSP0357
C                                                                       MCSP0358
      DO 48 I=1,NR                                                      CH200016
      DO 48 K=1,NSAM                                                    MCSP0359
C                                                                       MCSP0360
      DC 48 J=1,N                                                       MCSP0361
```

```
      XM(I,J,K) = 0.                                                    CH200017
      ERR(I,J,K) = 0.                                                   CH200019
C                                                                       MCSP0364
      DO 48 L=1,N                                                       MCSP0365
   48 VAR(J,L,K) = 0.                                                   MCSP0366
C                                                                       MCSP0367
C                                                                       MCSP0368
C                                                                       MCSP0369
C     BEGIN MAIN ITERATION LOOP HERE                                    MCSP0370
C                                                                       MCSP0371
      DC 54 ITER=1,NENS                                                 MCSP0372
C                                                                       MCSP0375
   49 DO 50 I=1,N                                                       MCSP0376
   50 XHKKM1(I) = XHATZ(1,I)                                            CH200018
C                                                                       MCSP0378
C                                                                       MCSP0379
      DO 54 K=1,NSAM                                                    MCSP0380
C     FORM NOISY MEASUREMENT FROM TRUE STATE VALUE                      MCSP0381
C                                                                       MCSP0382
C                                                                       MCSP0383
      DO 51 I=1,N                                                       MCSP0384
   51 X(I) = XS(1,I,K)                                                  CH200020
C                                                                       MCSP0386
      CALL GAIN                                                         MCSP0396
C                                                                       MCSP0397
      DO 52 I=1,N                                                       MCSP0398
      DO 52 J=1,M                                                       MCSP0400
   52 GKS(I,J,K) = G(I,J)                                               MCSP0401
C                                                                       MCSP0402
C                                                                       MCSP0403
C     UPCATE THE STATE ESTIMATE                                         MCSP0404
C                                                                       MCSP0405
   53 CALL ESTIM                                                        MCSP0406
C                                                                       MCSP0407
C     UPDATE RUNNING SUMS USED IN COMPUTING STATISTICS                  MCSP0408
      CALL STAT                                                         MCSP0409
C                                                                       MCSP0411
   54 CCNTINUE                                                          MCSP0414
C                                                                       MCSP0415
C                                                                       MCSP0416
C     DIVIDE RUNNING SUMS COMPUTED BY SUBROUTINE STAT BY ENSEMBLE       MCSP0417
C     SIZE TO COMPUTE STATISTICS                                        MCSP0418
      ENS = NENS                                                        MCSP0419
C                                                                       MCSP0420
      DO 56 K=1,NSAM                                                    MCSP0421
C                                                                       MCSP0422
      DO 56 J=1,N                                                       MCSP0423
   55 ERR(1,J,K) = ERR(1,J,K)/ENS                                       CH200022
```

```
   56 VAR(J,J,K) = VAR(J,J,K)/ENS-ERR(1,J,K)**2                             CH2*****
C
C
C                                                                               MCSP0428
C                                                                               MCSP0444
      IF (IPRT.NE.0) GO TO 64                                                   MCSP0445
      CALL PRT
   64 IF (IPLT.NE.0) GO TO 80                                                   MCSP0489
C                                                                               MCSP0490
      CALL PLT
   80 CONTINUE                                                                  MCSP0552
      STOP                                                                      MCSP0553
C                                                                               MCSP0554
   81 FORMAT (6(I10))                                                           CH200024
   82 FORMAT (I2)                                                               MCSP0556
   83 FORMAT (7(I10))                                                           MCSP0557
   84 FORMAT (2(I5))                                                            MCSP0558
   85 FORMAT (5(I10))
  130 FORMAT (4X,'N=',I2,4X,'M=',I2,4X'IN=',I2,4X'NSAM=',I3,4X,'NENS=',IMCSP0616
     15,4X'ND=',I2,//)                                                          MCSP0617
  131 FORMAT (//,10X,'THE PHI MATRIX IS',/)                                     MCSP0618
  132 FORMAT (//,10X,'THE H MATRIX IS',/)                                       MCSP0619
  133 FORMAT (//,10X,'THE R MATRIX IS',/)                                       MCSP0620
  134 FORMAT (//,10X,'THE COVARIANCE OF W MATRIX IS',/)                         MCSP0621
  135 FORMAT (//,10X,'THE Q MATRIX IS',/)                                       MCSP0622
  136 FORMAT (//,10X,'THE GAMMA MATRIX IS',/)                                   MCSP0623
  137 FORMAT (//,10X,'THE MATRIX P(0/-1) IS',/)                                 MCSP0624
  138 FORMAT (//,10X,'THE STD. DEVIATIONS OF MEASUREMENT NOISE ARE',/)          MCSP0625
  139 FORMAT (//,10X,'THE STD. DEVIATIONS OF INPUT FORCING W ARE',/)            MCSP0626
  140 FORMAT (//,10X,'THE VECTOR XHAT(0/-1) IS',/)                              MCSP0627
  141 FORMAT (//,10X,'THE MEAN OF THE VECTOR X(0) IS',/)                        MCSP0628
  142 FORMAT (//,10X,'THE STANDARD DEVIATIONS OF THE VECTOR X(0) ARE',/)MCSP0629
  143 FORMAT (//,10X,'THE INITIAL STATE IS',/)                                  MCSP0630
  144 FORMAT (4F20.0)                                                           MCSP0631
  145 FORMAT (//,10X,'THE FIRST AND LAST POINTS ON THE SINGLE TRACK TO BMCSP0632
     1E USED ARE',/)                                                            MCSP0633
  146 FORMAT (9(2X,1PE12.5),/)
  156 FORMAT ('1')
      END                                                                       MCSP0654
C                                                                               MCSP0399
      SUBROUTINE QMAT                                                           MCSP0764
C                                                                               MCSP0765
C     THIS SUBROUTINE COMPUTES THE MATRIX Q FROM THE EQUATION                   MCSP0766
C                                                                               MCSP0767
C        Q=GAMMA* E(W*WT) * GAMMAT                                              MCSP0768
C                                                                               MCSP0769
C     DOUBLE PRECISION ARITHMETIC IS USED                                       MCSP0770
C                                                                               MCSP0771
```

```
C                                                                       MCSP0772
      REAL*8 GAMMA,COVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,G,PKK,Q,EI,PR    CH3*****
      COMMON EI(4,4),Q(4,4),G(4,4),PKK(4,4),GAMMA(4,4),COVW(4,4),       MCSP0774
     1TEMP(4,4),TEMP1(4,4),TEMP2(4,4),H(4,4),PKKM1(4,4),R(4,4),PHI(4,4),MCSP0775
     2VAR(4,4,60),GKS(4,4,60),PKKS(4,4,60),XM(4,4,60),ERR(4,4,60),     CH200025
     3GAMMAS(4,4),PHIS(4,4),XS(4,4,60),HS(4,4),GK(4,4),SIGW(4),X(4),   CH200026
     4SIGXZ(4),XZMEAN(4),XHKK(4),XHKKM1(4),VTMP(4),Z(4),V(4),SIGV(4),   MCSP0778
     5XHATZ(4,4),XZ(60),YZ(60),PX(10),PY(10),PR(4,4,4),                 CH200004
     6N,NSAM,IQ,M,ITER,ITRK,IN,ISTAT,K,ITRO,IXZ,IV,IW,IEST,ND,NR        CH2
C                                                                       MCSP0781
C                                                                       MCSP0782
      CALL PROD (GAMMA,COVW,N,IN,IN,TEMP)                               MCSP0783
      CALL TRANS (GAMMA,N,IN,TEMP1)                                     MCSP0784
      CALL PROD (TEMP,TEMP1,N,IN,N,Q)                                   MCSP0785
      RETURN                                                            MCSP0786
      END                                                               MCSP0787
      SUBROUTINE QON                                                    MCSP0788
C                                                                       MCSP0789
C     IF Q IS TO BE COMPUTED ON-LINE (IFLQ.NE.0) IT IS DONE             MCSP0790
C     IN THIS SUBROUTINE                                                MCSP0791
C                                                                       MCSP0792
      REAL*8 GAMMA,COVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,G,PKK,Q,EI,PR    CH3*****
      COMMON EI(4,4),Q(4,4),G(4,4),PKK(4,4),GAMMA(4,4),COVW(4,4),       MCSP0794
     1TEMP(4,4),TEMP1(4,4),TEMP2(4,4),H(4,4),PKKM1(4,4),R(4,4),PHI(4,4),MCSP0795
     2VAR(4,4,60),GKS(4,4,60),PKKS(4,4,60),XM(4,4,60),ERR(4,4,60),     CH200028
     3GAMMAS(4,4),PHIS(4,4),XS(4,4,60),HS(4,4),GK(4,4),SIGW(4),X(4),   CH200029
     4SIGXZ(4),XZMEAN(4),XHKK(4),XHKKM1(4),VTMP(4),Z(4),V(4),SIGV(4),   MCSP0798
     5XHATZ(4,4),XZ(60),YZ(60),PX(10),PY(10),PR(4,4,4),                 CH200004
     6N,NSAM,IQ,M,ITER,ITRK,IN,ISTAT,K,ITRO,IXZ,IV,IW,IEST,ND,NR        CH2
C                                                                       MCSP0801
C     THE APPROPRIATE STATEMENTS FOR COMPUTING Q ON-LINE MUST           MCSP0802
C     BE INSERTED HERE BY THE USER                                      MCSP0803
C                                                                       MCSP0804
      RETURN                                                            MCSP0805
      END                                                               MCSP0806
      SUBROUTINE RON                                                    MCSP0807
C                                                                       MCSP0808
C     IF R IS TO BE COMPUTED ON-LINE (IFLR.NE.0) IT IS DONE             MCSP0809
C     IN THIS SUBROUTINE                                                MCSP0810
C                                                                       MCSP0811
      REAL*8 GAMMA,COVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,G,PKK,Q,EI,PR    CH3*****
      COMMON EI(4,4),Q(4,4),G(4,4),PKK(4,4),GAMMA(4,4),COVW(4,4),       MCSP0813
     1TEMP(4,4),TEMP1(4,4),TEMP2(4,4),H(4,4),PKKM1(4,4),R(4,4),PHI(4,4),MCSP0814
     2VAR(4,4,60),GKS(4,4,60),PKKS(4,4,60),XM(4,4,60),ERR(4,4,60),     CH200031
     3GAMMAS(4,4),PHIS(4,4),XS(4,4,60),HS(4,4),GK(4,4),SIGW(4),X(4),   CH200032
     4SIGXZ(4),XZMEAN(4),XHKK(4),XHKKM1(4),VTMP(4),Z(4),V(4),SIGV(4),   MCSP0817
     5XHATZ(4,4),XZ(60),YZ(60),PX(10),PY(10),PR(4,4,4),                 CH200004
     6N,NSAM,IQ,M,ITER,ITRK,IN,ISTAT,K,ITRO,IXZ,IV,IW,IEST,ND,NR        CH2
```

```
C                                                                       MCSP0820
C     THE APPROPRIATE STATEMENTS FOR COMPUTING R ON-LINE MUST           MCSP0821
C     BE INSERTED HERE BY THE USER                                      MCSP0822
C                                                                       MCSP0823
      RETURN                                                            MCSP0824
      END                                                               MCSP0825
C                                                                       MCSP0394
      SLBROUTINE STAT                                                   MCSP0826
C                                                                       MCSP0315
C     THIS SUBROUTINE  COMPUTES RUNNING SUMS USED IN DETERMINING THE    MCSP0827
C     SAMPLE STATISTICS OF TRACK AND ESTIMATION ERRORS. IN THE DEFAULT  MCSP0828
C     OPTION (ISTAT.EQ.0) THE STATISTICS TO BE COMPUTED ARE MEAN OF     MCSP0829
C     TRACK, MEAN OF ESTIMATION ERROR AND VARIANCE OF ESTIMATION        MCSP0830
C     ERROR. IF(ISTAT.NE.0) THE OFF-DIAGONAL TERMS IN THE COVARIANCE OF MCSP0831
C     ESTIMATION ERROR MATRIX ARE ALSO COMPUTED.                        MCSP0832
      REAL*8 GAMMA,COVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,G,PKK,Q,EI,PR    CH3*****
      COMMON EI(4,4),Q(4,4),G(4,4),PKK(4,4),GAMMA(4,4),COVW(4,4),       MCSP0834
     1TEMP(4,4),TEMP1(4,4),TEMP2(4,4),H(4,4),PKKM1(4,4),R(4,4),PHI(4,4),MCSP0835
     2VAR(4,4,60),GKS(4,4,60),PKKS(4,4,60),XM(4,4,60),ERR(4,4,60),      CH200034
     3GAMMAS(4,4),PHIS(4,4),XS(4,4,60),HS(4,4),GK(4,4),SIGW(4),X(4),    CH200035
     4SIGXZ(4),XZMEAN(4),XHKK(4),XHKKM1(4),VTMP(4),Z(4),V(4),SIGV(4),   MCSP0838
     5XHATZ(4,4),XZ(60),YZ(60),PX(10),PY(10),PR(4,4,4),                 CH200004
     6N,NSAM,IQ,M,ITER,ITRK,IN,ISTAT,K,ITRO,IXZ,IV,IW,IEST,ND,NR        CH2
      DIMENSION EXH(3)                                                  MCSP0841
      IF (ITRK.NE.1) GO TO 2                                            MCSP0842
      IF (ITER.NE.1) GO TO 4                                            MCSP0843
C                                                                       MCSP0844
      DO 1 J=1,N                                                        MCSP0845
    1 XM(1,J,K) = XS(1,J,K)                                             CH200037
C                                                                       MCSP0847
      GC TO 4                                                           MCSP0848
    2 CONTINUE                                                          MCSP0849
C                                                                       MCSP0850
      DC 3 J=1,N                                                        MCSP0851
    3 XM(1,J,K) =XM(1,J,K)+XS(1,J,K)                                    CH200038
C                                                                       MCSP0853
    4 CCNTINUE                                                          MCSP0854
C                                                                       MCSP0855
      DO 5 J=1,N                                                        MCSP0856
      EXH(J) = XHKK(J)-XS(1,J,K)                                        CH200039
      ERR(1,J,K)  = ERR(1,J,K)+EXH(J)                                   CH2
    5 VAR(J,J,K) = VAR(J,J,K)+EXH(J)**2                                 MCSP0859
C                                                                       MCSP0860
      IF (ISTAT.EQ.0) RETURN                                            MCSP0861
C                                                                       MCSP0862
      DC 6 L=2,N                                                        MCSP0863
      LM1 = L-1                                                         MCSP0864
C                                                                       MCSP0865
```

```
      DO 6 J=1,LM1                                                        MCSP0866
    6 VAR(L,J,K) = VAR(L,J,K)+EXH(L)*EXH(J)                               MCSP0867
C                                                                         MCSP0868
      RETURN                                                              MCSP0869
      END                                                                 MCSP0870
      SUBROUTINE XZERO                                                    MCSP0934
C     THIS SUBROUTINE GENERATES THE INITIAL STATE VALUE FROM A NORMAL     MCSP0935
C     RANDOM NUMBER GENERATOR. IT IS ASSUMED THAT THE INITIAL STATE       MCSP0936
C     HAS COMPONENTS THAT ARE INDEPENDENT                                 MCSP0937
      REAL*8 GAMMA,COVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,G,PKK,Q,EI,PR      CH3*****
      COMMON EI(4,4),Q(4,4),G(4,4),PKK(4,4),GAMMA(4,4),COVW(4,4),         MCSP0939
     1TEMP(4,4),TEMP1(4,4),TEMP2(4,4),H(4,4),PKKM1(4,4),R(4,4),PHI(4,4),MCSP0940
     2VAR(4,4,60),GKS(4,4,60),PKKS(4,4,60),XM(4,4,60),ERR(4,4,60),        CH200040
     3GAMMAS(4,4),PHIS(4,4),XS(4,4,60),HS(4,4),GK(4,4),SIGW(4),X(4),      CH200041
     4SIGXZ(4),XZMEAN(4),XHKK(4),XHKKM1(4),VTMP(4),Z(4),V(4),SIGV(4),     MCSP0943
     5XHATZ(4,4),XZ(60),YZ(60),PX(10),PY(10),PR(4,4,4),                  CH20000*
     6N,NSAM,IQ,M,ITER,ITRK,IN,ISTAT,K,ITRO,IXZ,IV,IW,IEST,ND,NR          CH2
      CALL SNORM (IXZ,X,N)                                                MCSP0946
C                                                                         MCSP0947
      DO 1 I=1,N                                                          MCSP0948
    1 XS(1,I,1) = SIGXZ(I)*X(I)+XZMEAN(I)                                 CH2
C                                                                         MCSP0950
      RETURN                                                              MCSP0951
      END                                                                 MCSP0952
      SUBROUTINE ADD (A,B,N,M,C)                                          MCSP0953
C     THIS SUBROUTINE ADDS THE NXM MATRICES A AND B, STORING THE          MCSP0954
C     RESULT IN C                                                         MCSP0955
      REAL*8 A,B,C                                                        MCSP0956
      DIMENSION A(4,4),B(4,4),C(4,4)                                      MCSP0957
C                                                                         MCSP0958
      DO 1 I=1,N                                                          MCSP0959
C                                                                         MCSP0960
      DO 1 J=1,M                                                          MCSP0961
    1 C(I,J) = A(I,J)+B(I,J)                                              MCSP0962
C                                                                         MCSP0963
      RETURN                                                              MCSP0964
      END                                                                 MCSP0965
      SUBROUTINE MREAD (A,N,M)                                            MCSP0966
C     8D10.5. THE ENTRIES IN THE FIRST ROW OF A ARE READ FIRST, THEN      MCSP0968
C     THIS SUBROUTINE READS AN NXM MATRIX A ACCORDING TO THE FORMAT       MCSP0967
C     THE ENTRIES IN THE SECOND ROW, AND SO ON.                           MCSP0969
      REAL*8 A                                                            MCSP0970
      DIMENSION A(4,4)                                                    MCSP0971
C                                                                         MCSP0972
      DO 1 I=1,N                                                          MCSP0973
    1 READ (5,2) (A(I,J),J=1,M)                                           MCSP0974
C                                                                         MCSP0975
      RETURN                                                              MCSP0976
```

```
C                                                                       MCSP0977
    2 FORMAT (8F10.0)                                                   MCSP0978
      END                                                               MCSP0979
      SLBROUTINE MWRITE (A,N,M)                                         MCSP0980
C     THIS SUBROUTINE WRITES THE ENTRIES OF THE NXM MATRIX A            MCSP0981
      REAL*8 A                                                          MCSP0982
      DIMENSION A(4,4)                                                  MCSP0983
C                                                                       MCSP0984
      DC 1 I=1,N                                                        MCSP0985
    1 WRITE (6,2) (A(I,J),J=1,M)                                        MCSP0986
C                                                                       MCSP0987
      RETURN                                                            MCSP0988
C                                                                       MCSP0989
    2 FORMAT (9(2X,1PE12.5))                                            MCSP0990
      END                                                               MCSP0991
      SLBROUTINE PROD (A,B,N,M,L,C)                                     MCSP0992
C     THIS SUBROUTINE COMPUTES THE MATRIX PRODUCT AB AND STORES THE     MCSP0993
C     RESULT IN C                                                       MCSP0994
C     A = NXM,   B = MXL,   C = NXL                                     MCSP0995
      REAL*8 A,B,C,T                                                    MCSP0996
      DIPENSION A(4,4),B(4,4),C(4,4),T(4,4)                             MCSP0997
C                                                                       MCSP0998
      DC 1 I=1,N                                                        MCSP0999
C                                                                       MCSP1000
      DO 1 J=1,L                                                        MCSP1001
    1 T(I,J) = 0.0                                                      MCSP1002
C                                                                       MCSP1003
C                                                                       MCSP1004
      DO 2 I=1,N                                                        MCSP1005
C                                                                       MCSP1006
      DO 2 J=1,L                                                        MCSP1007
C                                                                       MCSP1008
      DO 2 K=1,M                                                        MCSP1009
    2 T(I,J) = T(I,J)+A(I,K)*B(K,J)                                     MCSP1010
C                                                                       MCSP1011
C                                                                       MCSP1012
      DO 3 I=1,N                                                        MCSP1013
C                                                                       MCSP1014
      DO 3 J=1,L                                                        MCSP1015
    3 C(I,J) = T(I,J)                                                   MCSP1016
C                                                                       MCSP1017
      RETURN                                                            MCSP1018
      END                                                               MCSP1019
      SUBROUTINE SUB (A,B,N,M,C)                                        MCSP1020
C     THIS SUBROUTINE SUBTRACTS THE NXM MATRIX B FROM THE NXM MATRIX    MCSP1021
C     A AND STORES THE RESULT IN C                                      MCSP1022
      REAL*8 A,B,C                                                      MCSP1023
      DIMENSION A(4,4),B(4,4),C(4,4)                                    MCSP1024
```

```
C                                                                       MCSP1025
      DC 1 I=1,N                                                        MCSP1026
C                                                                       MCSP1027
      DO 1 J=1,M                                                        MCSP1028
    1 C(I,J) = A(I,J)-B(I,J)                                            MCSP1029
C                                                                       MCSP1030
      RETURN                                                            MCSP1031
      END                                                               MCSP1032
      SUBROUTINE TRANS (A,N,M,C)                                        MCSP1033
C     THIS SUBROUTINE FORMS THE MATRIX TRANSPOSE CF A STORING THE       MCSP1034
C     RESULT IN C                                                       MCSP1035
C     A = NXM,    C = MXN                                               MCSP1036
      REAL*8 A,C                                                        MCSP1037
      DIMENSION A(4,4),C(4,4)                                           MCSP1038
C                                                                       MCSP1039
      DO 1 I=1,N                                                        MCSP1040
C                                                                       MCSP1041
      DC 1 J=1,M                                                        MCSP1042
    1 C(J,I) = A(I,J)                                                   MCSP1043
C                                                                       MCSP1044
      RETURN                                                            MCSP1045
      END                                                               MCSP1046
      SLBROUTINE VADD (X,Y,N,Z)                                         MCSP1047
C     THIS SUBROUTINE COMPUTES THE SUM OF THE N-VECTORS X AND           MCSP1048
C     Y AND STORES THE RESULT IN THE N-VECTOR Z                         MCSP1049
C                                                                       MCSP1050
      REAL*4 X(4),Y(4),Z(4)                                             MCSP1051
C                                                                       MCSP1052
      DO 1 I=1,N                                                        MCSP1053
    1 Z(I) = X(I)+Y(I)                                                  MCSP1054
C                                                                       MCSP1055
      RETURN                                                            MCSP1056
      END                                                               MCSP1057
      SUBROUTINE VPROD (A,X,M,N,Y)                                      MCSP1058
C     THIS SUBROUTINE COMPUTES THE PRODUCT OF THE MXN MATRIX            MCSP1059
C     A AND TFE N-VECTOR X AND STORES THE RESULT IN THE                 MCSP1060
C     M-VECTOR Y                                                        MCSP1061
C                                                                       MCSP1062
C                                                                       MCSP1063
      REAL*4 A(4,4),X(4),Y(4),T(4)                                      MCSP1064
C                                                                       MCSP1065
      DO 1 I=1,M                                                        MCSP1066
      T(I) = 0.D0                                                       MCSP1067
C                                                                       MCSP1068
      DO 1 J=1,N                                                        MCSP1069
    1 T(I) = T(I)+A(I,J)*X(J)                                           MCSP1070
C                                                                       MCSP1071
C                                                                       MCSP1072
```

```
      DO 2 I=1,M                                                        MCSP1073
    2 Y(1) = T(I)                                                       MCSP1074
C                                                                       MCSP1075
C
      RETURN                                                            MCSP1076
      END                                                               MCSP1077
      SLBROUTINE VREAD (V,N)                                            MCSP1078
C                                                                       MCSP1079
C     THIS SUBROUTINE READS THE N-DIMENSIONAL S.P. VECTOR V             MCSP1080
C                                                                       MCSP1081
      DIMENSION V(4)                                                    MCSP1082
      READ (5,1) (V(I),I=1,N)                                           MCSP1083
      RETURN
    1 FORMAT (8F10.0)
      END
      SUBROUTINE VSUB (X,Y,N,Z)
      REAL*4 X(4),Y(4),Z(4)
      DO 1 I=1,N
    1 Z(I) = X(I)-Y(I)
      RETURN
      END
      SUBROUTINE VWRITE (V,N)
      DIMENSICN V(4)
      WRITE (6,1) (V(I),I=1,N)
      RETURN
    1 FCRMAT (9(2X,1PE12.5))
      END
      SLBROUTINE TRACK                                                  MCSP0871
C     IF TRACK IS TO BE GENERATED ON-LINE IT IS DCNE IN THIS SUBROUTINE MCSP0872
C     IN THE DEFAULT OPTION (ITRK.EQ.0) THE TRACK IS GENERATED          MCSP0873
C     FROM THE STANDARD LINEAR DIFFERENCE EQUATION                      MCSP0874
C                                                                       MCSP0875
C     X(K+1)=PHI*X(K)+GAMMA*W(K)                                        MCSP0876
C                                                                       MCSP0877
      REAL*8 GAMMA,COVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,G,PKK,Q,EI,PR    CH3*****
      CCMMON EI(4,4),Q(4,4),G(4,4),PKK(4,4),GAMMA(4,4),COVW(4,4),       MCSP0879
     1TEMP(4,4),TEMP1(4,4),TEMP2(4,4),H(4,4),PKKM1(4,4),R(4,4),PHI(4,4),MCSP0666
     2VAR(4,4,60),GKS(4,4,60),PKKS(4,4,60),XM(4,4,60),ERR(4,4,60),      CH200043
     3GAMMAS(4,4),PHIS(4,4),XS(4,4,60),HS(4,4),GK(4,4),SIGW(4),X(4),    CH200044
     4SIGXZ(4),XZMEAN(4),XHKK(4),XHKKM1(4),VTMP(4),Z(4),V(4),SIGV(4),   MCSP0883
     5XHATZ(4,4),XZ(60),YZ(60),PX(10),PY(10),PR(4,4,4),                 CH20000*
     6N,NSAM,IQ,M,ITER,ITRK,IN,ISTAT,K,ITRO,IXZ,IV,IW,IEST,ND,NR        CH2
      DIMENSION W(3)                                                    MCSP0886
C     TO GENERATE A SINGLE TRAJECTORY AND STORE IT IN THE ARRAY         MCSP0913
C     XS(I,K),I=1,N,K=2,NSAM (NOTE THAT IF A SINGLE TRAJECTORY IS TO BE MCSP0914
C     GENERATED, THE INITIAL CONDITION HAS BEEN READ IN AND STORED      MCSP0915
C     IN XS(I,1),I=1,N)                                                 MCSP0916
C                                                                       MCSP0917
```

```
      TPI = 2.*3.14159265
      DC 5 K=2,NSAM                                                     MCSP0918
      EKM1 = K-1                                                        MCSP0919
      T = 1.0*EKM1
      A=0.03333*T
      IF(A.LT.TPI) GO TO 10
      MM= A/TPI
      FM=MM
      A = A - FM*TPI
   10 CCNTINUE
      XS(1,1,K)=10.*SIN(A)                                              CH200046
      XS(1,2,K)=.3333*COS(A)                                            CH200047
      XS(1,3,K)=10.*COS(A)                                              CH200048
      XS(1,4,K)=-.3333*SIN(A)                                           CH200049
      DO 7 I=2,NR                                                       CH200050
      EKM1 = K-1                                                        CH200051
      XS(I,1,K) = XS(I,1,EKM1) + XS(I,2,1)                              CH200053
      XS(I,2,K) = XS(I,2,1)                                             CH200054
      XS(I,3,K) = XS(I,3,EKM1) + XS(I,4,1)                              CH200055
    7 XS(I,4,K) = XS(I,4,1)                                             CH200056
    5 CCNTINUE                                                          CH200057
C                                                                       MCSP0925
      RETURN                                                            MCSP0926
      END                                                               MCSP0933
      SLBROUTINE MEAS                                                   MCSP0734
C     THIS SUBROUTINE STARTS WITH THE TRUE STATE VALUE XS               MCSP0735
C     AND ADDS ZERO-MEAN WHITE GAUSSIAN NOISE TO H*XS TO                MCSP0736
C     GENERATE A NOISY VECTOR OF MEASUREMENTS Z.                        MCSP0737
C                                                                       MCSP0738
C                                                                       MCSP0739
      REAL*8 GAMMA,COVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,G,PKK,Q,EI,PR    CH3*****
      COMMON EI(4,4),Q(4,4),G(4,4),PKK(4,4),GAMMA(4,4),COVW(4,4),       MCSP0741
     1TEMP(4,4),TEMP1(4,4),TEMP2(4,4),H(4,4),PKKM1(4,4),R(4,4),PHI(4,4),MCSP0742
     2VAR(4,4,60),GKS(4,4,60),PKKS(4,4,60),XM(4,4,60),ERR(4,4,60),      CH200058
     3GAMMAS(4,4),PHIS(4,4),XS(4,4,60),HS(4,4),GK(4,4),SIGW(4),X(4),    CH200059
     4SIGXZ(4),XZMEAN(4),XHKK(4),XHKKM1(4),VTMP(4),Z(4),V(4),SIGV(4),   MCSP0745
     5XHATZ(4,4),XZ(60),YZ(60),PX(10),PY(10),PR(4,4,4),                 CH20000*
     6N,NSAM,IQ,M,ITER,ITRK,IN,ISTAT,K,ITRO,IXZ,IV,IW,IEST,ND,NR        CH2
      ALPHA = XS(1,3,K)                                                 CH200061
      BETA = XS(1,1,K)                                                  CH200062
      Z(1) = SQRT(ALPHA**2+BETA**2)                                     MCSP0750
      Z(2) = ATAN2(ALPHA,BETA)                                          MCSP0751
      CALL SNORM (IV,V,M)                                               MCSP0752
C                                                                       MCSP0753
      DC 1 I=1,M                                                        MCSP0754
    1 V(I) = SIGV(I)*V(I)                                               MCSP0755
C                                                                       MCSP0756
      CALL VADD (Z,V,M,Z)                                               MCSP0757
```

```
      ALPHA = Z(1)*COS(Z(2))                                            MCSP0758
      BETA = Z(1)*SIN(Z(2))                                             MCSP0759
      XZ(K)=ALPHA
      YZ(K)=BETA
      RETURN                                                            MCSP0762
      END                                                               MCSP0763
C                                                                       MCSP0216
      SLBROUTINE GAIN                                                   MCSP0695
C                                                                       MCSP0218
      REAL*8 GAMMA,COVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,G,PKK,Q,EI,PR    CH3*****
      COMMON EI(4,4),Q(4,4),G(4,4),PKK(4,4),GAMMA(4,4),COVW(4,4),       MCSP0697
     1TEMP(4,4),TEMP1(4,4),TEMP2(4,4),H(4,4),PKKM1(4,4),R(4,4),PHI(4,4),MCSP0698
     2VAR(4,4,60),GKS(4,4,60),PKKS(4,4,60),XM(4,4,60),ERR(4,4,60),      CH200063
     3GAMMAS(4,4),PHIS(4,4),XS(4,4,60),HS(4,4),GK(4,4),SIGW(4),X(4),    CH200064
     4SIGXZ(4),XZMEAN(4),XHKK(4),XHKKM1(4),VTMP(4),Z(4),V(4),SIGV(4),   MCSP0701
     5XHATZ(4,4),XZ(60),YZ(60),PX(10),PY(10),PR(4,4,4),                 CH20000*
     6N,NSAM,IQ,M,ITER,ITRK,IN,ISTAT,K,ITRO,IXZ,IV,IW,IEST,ND,NR        CH2
      DIMENSION BE(4),ER(4)
C                                                                       MCSP0704
C     G(K) = P(K/K-1)*HT*(H*P(K/K-1)*HT + R)                            MCSP0705
      DC 300 I=1,4
      DO 300 J=1,4
  300 PKKS(I,J,K)=PKKM1(I,J)
C                                                                       PLR05810
      IF(DABS(PKKM1(1,1)-PKKM1(3,3)).GT.0) GO TO 11
      PKKM1(1,1)=PKKM1(3,3)+0.000001
   11 CONTINUE
C                                                                       PLR05660
C     FINC UPDATE UNIT'S ELLIPSE ORIENTATION                            PLR05870
C                                                                       PLR05880
      THE=0.5*DATAN(2.*PKKM1(1,3)/(PKKM1(1,1)-PKKM1(3,3)))
      IF(ABS(THE).GT.0) GO TO 10
      TFE = 0.00001
   10 CONTINUE
C                                                                       PLR05930
C     FINE UNCOUPLED VARIANCES                                          PLR05940
C                                                                       PLR05950
      SIG2X=(PKKM1(1,1)+PKKM1(3,3))/2.+PKKM1(1,3)/SIN(2.*THE)
      SIG2Y=(PKKM1(1,1)+PKKM1(3,3))/2.-PKKM1(1,3)/SIN(2.*THE)
C                                                                       PLR05980
C     ACJUST THETA                                                      PLR05990
C                                                                       PLR06000
      IF(SIG2X.GE.SIG2Y) GO TO 63
      TFE=THE+3.14159265/2.
C
C     CALCULATE BEARING
C
      DC 9 IN=1,3                                                       CH200066
```

```
      IO = IN + 1                                                       CH200067
      IF(ABS(XHKKM1(1)-XS(IO,1,K)).GT.0) GO TO 9                        CH200068
      XHKKM1(1)= 0.0000001+XS(IO,1,K)                                   CH200069
    9 BE(IN)=ATAN((XHKKM1(3)-XS(IO,3,K))/(XHKKM1(1)-XS(IO,1,K)))        CH200070
   63 CONTINUE
   66 DO 4 IN=1,3
    4 ER(IN)= ABS(THE-BE(IN))
C
C     CHCOSE BEST RANGER
C
      IF(ABS(COS(ER(1))).LE.ABS(COS(ER(2)))) GO TO 7
      IF(ABS(COS(ER(1))).LE.ABS(COS(ER(3)))) GO TO 70
      IN=1
      GC TO 8
    7 IF(ABS(COS(ER(2))).LE.ABS(COS(ER(3))))GO TO 70
      IN=2
      GC TO 8
   70 IN=3
    8 XIN=IN
      IC=IN +1
C                                                                       PLR06200
C     CALCULATE H                                                       PLR06210
C                                                                       PLR06220
      RR =((XHKKM1(1)-XS(IO,1,K))**2+(XHKKM1(3)-XS(IO,3,K))**2)**.5     CH200071
      WRITE (6,22)                                                      CH3*****
   22 FORMAT(/,6X,'THE',12X,'SIG2X',8X,'SIG2Y',10X,'BE(1)', 8X,'BE(2)', CH3*****
     18X,'XIN',10X,'ER(1)',9X,'ER(2)',10X,'RR')                         CH3*****
      WRITE (6,146) THE,SIG2X,SIG2Y,BE(1),BE(2),XIN,ER(1),ER(2), RR
  146 FORMAT (9(2X,1PE12.5),/)                                          MCSP0634
      H(1,1)=(XHKKM1(1)-XS(IO,1,K))/RR                                  CH200072
      H(1,3) = (XHKKM1(3)-XS(IO,3,K))/RR                                CH200073
      IF(DABS(PR(1,1,IN)-PR(3,3,IN)).GT.0) GO TO 20                     CH3*****
      PR(3,3,IN) = PR(1,1,IN)+0.000001                                  CH3*****
   20 CONTINUE                                                          CH3*****
C                                                                       PLR06300
C     FIND RANGER'S ERROR ELLIPSE ORIENTATION (THER)                    PLR06310
C
      THER = 0.5*DATAN(2.*PR(1,3,IN)/(PR(1,1,IN)-PR(3,3,IN)))           CH3****7
      IF(ABS(THER).GT.0) GO TO 19                                       CH3*****
      THER = 0.00001                                                    CH3*****
   19 SIG2XR = (PR(1,1,IN) + PR(3,3,IN)) / 2. + PR(1,3,IN) / SIN(2.*THERCH3****8
     1)                                                                 CH*3***9
C
C     CALCULATE RANGER'S UNCORELATED VARIANCES
C
      SIG2YR = (PR(1,1,IN) + PR(3,3,IN)) / 2. - PR(1,3,IN) / SIN(2.*THERCH3***10
     1)                                                                 CH3***11
      IF(SIG2YR.GE.0.) GO TO 21                                         CH3?????
```

```
      SIG2YR = - SIG2YR                                                 CH3?????
   21 CONTINUE                                                          CH3?????
      ERP = BE(IN) - THER                                               CH3***12
C
C
C     CIND THE MAJOR AND MINOR AXES
      IF(SIG2XR.GE.SIG2YR)GO TO 24                                      PLR06430
      SIGMJ = SIG2YR                                                    PLR06440
      SIGMN = SIG2XR                                                    PLR06450
      GO TO 25                                                          PLR06460
   24 SIGMJ = SIG2XR                                                    PLR06470
      SIGMN = SIG2YR                                                    PLR06480
   25 CCNTINUE                                                          PLR06490
C
C     CALCULATE THE NOISE COVARIANCE
C
      R(1,1) = SIGMJ*SIGMN/ SIGMJ*((SIN(ERP))**2) + SIGMN*((COS(ERP))**2PLR06500
      WRITE (6,23)
   23 FCRMAT (6X,'THER',12X,'SIG2XR', 8X,'SIG2YR', 9X,'ERP',10X,'R(1,1)'CH3*****
     1,10X,'BE(3)')                                                     CH3*****
      WRITE(6,146)THER,SIG2XR,SIG2YR,ERP,R(1,1),BE(3)                   CH3*****
      CALL TRANS (H,M,N,TEMP2)                                          MCSP0706
      CALL PROD (PKKM1,TEMP2,N,N,M,TEMP)                                MCSP0707
      CALL PROD (H,TEMP,M,N,M,TEMP1)                                    MCSP0708
      CALL ADD (TEMP1,R,M,M,TEMP1)                                      MCSP0709
      IF (M.EQ.1) GO TO 2                                               MCSP0710
      MD = ND                                                           MCSP0711
      CALL GAUSS3 (M,EPS,TEMP1,TEMP2,KER,MD)                            MCSP0712
      CALL PROD (TEMP,TEMP2,N,M,M,G)                                    MCSP0713
C                                                                       MCSP0714
C     NOTE HERE  PKK(I,J) = P(K/K)  WHERE                               MCSP0715
C     P(K/K) = (I-G(K)*H)*P(K/K-1)                                      MCSP0716
    1 CALL PROD (G,H,N,M,N,TEMP)                                        MCSP0717
      CALL SUB (EI,TEMP,N,N,TEMP2)                                      MCSP0718
      CALL PROD (TEMP2,PKKM1,N,N,N,PKK)                                 MCSP0719
C                                                                       MCSP0720
C     NOTE HERE  PKKM1(I,J) = P(K/K-1)  WHERE                           MCSP0721
C     P(K/K-1)= PHI*P(K-1/K-1)*PHIT + Q                                 MCSP0722
      CALL TRANS (PHI,N,N,TEMP2)                                        MCSP0723
      CALL PROD (PKK,TEMP2,N,N,N,TEMP)                                  MCSP0724
      CALL PROD (PHI,TEMP,N,N,N,TEMP1)                                  MCSP0725
      CALL ADD (TEMP1,Q,N,N,PKKM1)                                      MCSP0726
      RETURN                                                            MCSP0727
C                                                                       MCSP0728
    2 DO 3 I=1,N                                                        MCSP0729
    3 G(I,1) = TEMP(I,1)/TEMP1(1,1)                                     MCSP0730
C                                                                       MCSP0731
      GO TO 1                                                           MCSP0732
```

```
      END                                                               MCSP0733
C                                                                       MCSP0206
      SLBROUTINE ESTIM                                                  MCSP0655
C                                                                       MCSP0209
C     THIS SUBROUTINE UPDATES THE STATE ESTIMATE. IN THE DEFAULT        MCSP0656
C     CONDITION (IEST.EQ.0) THE STANDARD EQUATIONS                      MCSP0657
C                                                                       MCSP0658
C     XHAT(K/)=XHAT(K/K-1)+G(K)*(Z(K)-H(K)*XHAT(K/K-1))                 MCSP0659
C                                                                       MCSP0660
C     XFAT(K+1/K)=PHI*XHAT(K/K)                                         MCSP0661
C                                                                       MCSP0662
C     ARE EVALUATED                                                     MCSP0663
      REAL*8 GAMMA,COVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,G,PKK,Q,EI,PR    CH3*****
      COMMON EI(4,4),Q(4,4),G(4,4),PKK(4,4),GAMMA(4,4),COVW(4,4),       MCSP0665
     1TEMP(4,4),TEMP1(4,4),TEMP2(4,4),H(4,4),PKKM1(4,4),R(4,4),PHI(4,4),MCSP0666
     2VAR(4,4,60),GKS(4,4,60),PKKS(4,4,60),XM(4,4,60),ERR(4,4,60),      CH200074
     3GAMMAS(4,4),PHIS(4,4),XS(4,4,60),HS(4,4),GK(4,4),SIGW(4),X(4),    CH200075
     4SIGXZ(4),XZMEAN(4),XHKK(4),XHKKM1(4),VTMP(4),Z(4),V(4),SIGV(4),   MCSP0669
     5XHATZ(4,4),XZ(60),YZ(60),PX(10),PY(10),PR(4,4,4),                 CH20000*
     6N,NSAM,IQ,M,ITER,ITRK,IN,ISTAT,K,ITRO,IXZ,IV,IW,IEST,ND,NR        CH2
C     TAKE THE APPROPRIATE GAIN AND STORE IN THE ARRAY GK               MCSP0672
C                                                                       MCSP0673
C                                                                       MCSP0675
      XIN=IN
      IC = IN + 1                                                       CH2
      DO 1 I=1,N                                                        MCSP0674
      DO 1 J=1,M                                                        MCSP0676
    1 GK(I,J) = GKS(I,J,K)                                              MCSP0677
C                                                                       MCSP0678
      CALL VPROD (HS,XHKKM1,M,N,VTMP)                                   MCSP0680
      VTMP(1)=((XHKKM1(1)-XS(IO,1,K))**2+(XHKKM1(3)-XS(IO,3,K))**2)**.5 CH200077
      Z(1) =  ((XZ(K)-XS(IO,1,K))**2+(YZ(K)-XS(IO,3,K))**2)**.5         CH200078
      CALL VSUB (Z,VTMP,M,VTMP)                                         MCSP0681
      CALL VPROD (GK,VTMP,N,M,VTMP)                                     MCSP0682
      CALL VADD (XHKKM1,VTMP,N,XHKK)                                    MCSP0683
C                                                                       MCSP0684
C     XHAT(K/K) HAS BEEN COMPUTED AND STORED IN THE ARRAY XHKK          MCSP0685
      CALL VPROD (PHIS,XHKK,N,N,XHKKM1)                                 MCSP0686
C
C     XHAT(K+1/K) HAS BEEN COMPUTED AND STORED IN THE ARRAY XHKKM1      MCSP0688
      RETURN                                                            MCSP0689
      END                                                               MCSP0694
      SUBROUTINE PRT
      REAL*8 GAMMA,COVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,G,PKK,Q,EI,PR    CH3*****
      COMMON EI(4,4),Q(4,4),G(4,4),PKK(4,4),GAMMA(4,4),COVW(4,4),       MCSP0879
     1TEMP(4,4),TEMP1(4,4),TEMP2(4,4),H(4,4),PKKM1(4,4),R(4,4),PHI(4,4),MCSP0666
     2VAR(4,4,60),GKS(4,4,60),PKKS(4,4,60),XM(4,4,60),ERR(4,4,60),      CH200079
     3GAMMAS(4,4),PHIS(4,4),XS(4,4,60),HS(4,4),GK(4,4),SIGW(4),X(4),    CH200080
```

```
     4SIGXZ(4),XZMEAN(4),XHKK(4),XHKKM1(4),VTMP(4),Z(4),V(4),SIGV(4),   MCSP0012
     5XHATZ(4,4),XZ(60),YZ(60),PX(10),PY(10),PR(4,4,4),                 CH20000*
     6N,NSAM,IQ,M,ITER,ITRK,IN,ISTAT,K,ITRO,IXZ,IV,IW,IEST,ND,NR        CH2
      WRITE (6,147)                                                     MCSP0446
C
C
C     WRITE GAINS, THEORETICAL COVARIANCES OF ESTIMATION ERROR          MCSP0447
      WRITE (6,148)                                                     MCSP0450
C                                                                       MCSP0451
      DC 59 K=1,NSAM                                                    MCSP0452
      WRITE (6,149) K                                                   MCSP0453
C                                                                       MCSP0454
      DO 59 I=1,N                                                       MCSP0455
   59 WRITE (6,146) (GKS(I,J,K),J=1,M)                                  MCSP0456
C                                                                       MCSP0457
      WRITE (6,150)                                                     MCSP0458
C                                                                       MCSP0459
      DO 60 K=1,NSAM                                                    MCSP0460
      WRITE (6,151) K                                                   MCSP0461
C                                                                       MCSP0462
      DO 60 I=1,N                                                       MCSP0463
   60 WRITE (6,146) (PKKS(I,J,K),J=1,N)                                 MCSP0464
C                                                                       MCSP0465
C                                                                       MCSP0466
C                                                                       MCSP0467
   61 WRITE (6,156)                                                     MCSP0468
      WRITE (6,152)                                                     MCSP0469
      WRITE (6,153)                                                     MCSP0470
C                                                                       MCSP0471
      DO 62 K=1,NSAM                                                    MCSP0472
      WRITE (6,155)                                                     MCSP0473
C                                                                       MCSP0474
      DO 62 I=1,N                                                       MCSP0475
   62 WRITE (6,154) K,I,XM(1,I,K),ERR(1,I,K),VAR(I,I,K)                 CH200082
C                                                                       MCSP0477
      WRITE (6,156)                                                     MCSP0478
C                                                                       MCSP0487
      WRITE (6,156)                                                     MCSP0488
  146 FORMAT (9(2X,1PE12.5),/)
  147 FCRMAT ('1',20X,'OUTPUT DATA',///)                                MCSP0635
  148 FORMAT (10X,'THE GAIN MATRICES ARE',/)                            MCSP0636
  149 FCRMAT (5X,'K=',I3,/,10X,'G(K)=',/)                               MCSP0637
  150 FORMAT (1X,//,10X,'THE THEORETICAL COVARIANCE MATRIX IS',/)       MCSP0638
  151 FORMAT (5X,'K=',I3,/,10X,'P(K/K)=',/)                             MCSP0639
  152 FCRMAT (T5,'TIME',T16,'VECTOR COM-',T34,'SAMPLE MEAN',            MCSP0640
     1 T51,'SAMPLE MEAN OF',T71,'SAMPLE VARIANCE CF')                   MCSP0641
  153 FORMAT (T5,'INDEX',T16,'PONENT INDEX',T34,'CF TRACK',             MCSP0642
     1 T51,'ESTIMATION ERROR',T71,'ESTIMATION ERROR')                   MCSP0643
```

```
  154 FORMAT (6X,I3,13X,I1,10X,1PE14.7,2(6X,1PE14.7))                    MCSP0644
  155 FORMAT (//)                                                          MCSP0645
  156 FORMAT ('1')                                                         MCSP0646
  157 FORMAT (10X,'THE SAMPLE COVARIANCE OF EST. ERROR MATRIX IS',//)      MCSP0647
  158 FORMAT (//,2X,'K=',I3,/)                                             MCSP0648
      RETURN
      END
      SUBROUTINE PLT
      REAL*8 GAMMA,COVW,R,PHI,H,TEMP,TEMP1,TEMP2,PKKM1,G,PKK,Q,EI,PR       CH3*****
      COMMON EI(4,4),Q(4,4),G(4,4),PKK(4,4),GAMMA(4,4),COVW(4,4),          MCSP0879
     1TEMP(4,4),TEMP1(4,4),TEMP2(4,4),H(4,4),PKKM1(4,4),R(4,4),PHI(4,4),MCSP0880
     2VAR(4,4,60),GKS(4,4,60),PKKS(4,4,60),XM(4,4,60),ERR(4,4,60),         CH200083
     3GAMMAS(4,4),PHIS(4,4),XS(4,4,60),HS(4,4),GK(4,4),SIGW(4),X(4),      CH200084
     4SIGXZ(4),XZMEAN(4),XHKK(4),XHKKM1(4),VTMP(4),Z(4),V(4),SIGV(4),     MCSP0883
     5XHATZ(4,4),XZ(60),YZ(60),PX(10),PY(10),PR(4,4,4),                    CH20000*
     6N,NSAM,IQ,M,ITER,ITRK,IN,ISTAT,K,ITRO,IXZ,IV,IW,IEST,ND,NR          CH2
      INTEGER*4 ITB(12)/12*0/
      REAL*4    RTB(28)/28*0.0/
      DIMENSION XP(60),YP(60)
      EQUIVALENCE (TITLE,RTB(5))
      REAL*8 TITLE(12)/'X = TRUE,  + = FILTER,  SQUARE = NOISY'/
C
C
      IGPLT=1
      ITHVPL=1
      INTPLT=1
      ISMPLT=1
      ISVPLT=1
      DO 500 KY=1,NR                                                       CH2*****
      KX=NR+1-KY                                                           CH2*****
      DO 50 K=1,NSAM
      XP(K) = XM(KX,1,K)                                                   CH2*****
   50 YP(K) = XM(KX,3,K)                                                   CH2*****
      CALL PLOTP(XP,YP,NSAM,0)
  500 CONTINUE                                                             CH2*****
      ITB(1)=1
      ITB(2)=1
      CALL DRAWP(60,XP,YP,ITB,RTB)
      DO 51 K=1,NSAM
      XP(K)=XS(1,1,K)+ERR(1,1,K)                                           CH200088
   51 YP(K)=XS(1,3,K)+ERR(1,3,K)                                           CH200089
      CALL PLOTP(XP,YP,NSAM,0)
      ITB(1)=2
      ITB(2)=2
      CALL DRAWP(60,XP,YP,ITB,RTB)
      ITB(2)=0
      DO 2 J=1,60,5
      IF(ABS(PKKS(1,1,J)-PKKS(3,3,J)).GT.0) GO TO 11
```

```
      PKKS(1,1,J)=PKKS(3,3,J)+0.000001
   11 CONTINUE
      THE=0.5*ATAN(2.*PKKS(1,3,J)/(PKKS(1,1,J)-PKKS(3,3,J)))
      IF(ABS(THE).GT.0) GO TO 10
      THE=0.000001
   10 CONTINUE
      SIG2X=(PKKS(1,1,J)+PKKS(3,3,J))/2.+PKKS(1,3,J)/SIN(2.*THE)
      SIG2Y=(PKKS(1,1,J)+PKKS(3,3,J))/2.-PKKS(1,3,J)/SIN(2.*THE)
      WRITE (6,146) THE,SIG2X,SIG2Y
  146 FORMAT (9(2X,1PE12.5),/)
      SX=(SIG2X)**.5*20.
      SY=(SIG2Y)**.5*20.
      PT=3.14159265/12.
      CT=COS( THE )
      ST=SIN( THE )
      DO 1 I=1,25
      XI=I
      XP(I)=SX*COS(PT*XI)*CT-SY*SIN(PT*XI)*ST+XS(1,1,J)                 CH200090
    1 YP(I)=SX*COS(PT*XI)*ST+SY*SIN(PT*XI)*CT+XS(1,3,J)                 CH200091
    2 CALL DRAWP(25,XP,YP,ITB,RTB)
      DO 201 J=2,NR                                                     CH200092
      DO 200 K=1,NSAM                                                   CH200093
      XP(K)=XS(J,1,K)                                                   CH200094
  200 YP(K)=XS(J,3,K)                                                   CH200095
      IT=(J/2)+3                                                        CH200096
      ITB(2)=IT                                                         CH200097
  201 CALL DRAWP (60,XP,YP,ITB,RTB)                                     CH200098
      ITB(1)=3
      ITB(2)=3
      CALL DRAWP(60,XZ,YZ,ITB,RTB)
      DO 65 K=1,NSAM                                                    MCSP0491
   65 XP(K) = K                                                         MCSP0492
C                                                                       MCSP0493
      IF (IGPLT.NE.1) GO TO 68                                          MCSP0494
C                                                                       MCSP0495
      DO 67 I=1,N                                                       MCSP0496
C                                                                       MCSP0497
      DO 67 J=1,M                                                       MCSP0498
C                                                                       MCSP0499
      DO 66 K=1,NSAM                                                    MCSP0500
   66 YP(K) = GKS(I,J,K)                                                MCSP0501
C                                                                       MCSP0502
      WRITE (6,156)                                                     MCSP0503
      CALL PLOTP (XP,YP,NSAM,0)                                         MCSP0504
   67 WRITE (6,159) I,J                                                 MCSP0505
C                                                                       MCSP0506
   68 IF (ITHVPL.NE.1) GO TO 71                                         MCSP0507
C                                                                       MCSP0508
```

```
      DO 70 I=1,N                                                       MCSP0509
C                                                                       MCSP0510
      DO 69 K=1,NSAM                                                    MCSP0511
   69 YP(K) = PKKS(I,I,K)                                               MCSP0512
C                                                                       MCSP0513
      WRITE (6,156)                                                     MCSP0514
      CALL PLOTP (XP,YP,NSAM,0)                                         MCSP0515
   70 WRITE (6,160) I,I                                                 MCSP0516
C                                                                       MCSP0517
   71 IF (IMTPLT.NE.1) GO TO 74                                         MCSP0518
C                                                                       MCSP0519
      DO 73 I=1,N                                                       MCSP0520
C                                                                       MCSP0521
      DO 72 K=1,NSAM                                                    MCSP0522
   72 YP(K) = XM(1,I,K)                                                 CH200099
C                                                                       MCSP0524
      WRITE (6,156)                                                     MCSP0525
      CALL PLOTP (XP,YP,NSAM,0)                                         MCSP0526
   73 WRITE (6,161) I                                                   MCSP0527
C                                                                       MCSP0528
   74 IF (ISMPLT.NE.1) GO TO 77                                         MCSP0529
C                                                                       MCSP0530
      DO 76 I=1,N                                                       MCSP0531
C                                                                       MCSP0532
      DO 75 K=1,NSAM                                                    MCSP0533
   75 YP(K) = ERR(1,I,K)                                                CH2
C                                                                       MCSP0535
      WRITE (6,156)                                                     MCSP0536
      CALL PLOTP (XP,YP,NSAM,0)                                         MCSP0537
   76 WRITE (6,162) I,I                                                 MCSP0538
C                                                                       MCSP0539
   77 IF (ISVPLT.NE.1) GO TO 80                                         MCSP0540
C                                                                       MCSP0541
      DO 79 I=1,N                                                       MCSP0542
C                                                                       MCSP0543
      DO 78 K=1,NSAM                                                    MCSP0544
   78 YP(K) = VAR(I,I,K)                                                MCSP0545
C                                                                       MCSP0546
      WRITE (6,156)                                                     MCSP0547
      CALL PLOTP (XP,YP,NSAM,0)                                         MCSP0548
   79 WRITE (6,163) I                                                   MCSP0549
   80 CONTINUE
C                                                                       MCSP0550
      WRITE (6,156)                                                     MCSP0551
  156 FORMAT ('1')
  159 FORMAT (12X,'G(',I1,',',I1,')  VS. K')                            MCSP0649
  160 FORMAT (12X,'PKK(',I1,',',I1,')  VS. K')                          MCSP0650
  161 FORMAT (12X,'MEAN OF X(',I1,')  VS. K')                           MCSP0651
```

```
  162 FORMAT (12X,'XHATKK(',I1,') -X(',I1,')  VS. K')                   MCSP0652
  163 FORMAT (12X,'ERROR VARIANCE(',I1,') VS.. K')                      MCSP0653
      RETURN
      END
//GO.FT06F001 DD SYSOUT=A,SPACE=(CYL,(4,1))
//GO.SYSIN DD *

$$$$$$$$$$DATA DECK$$$$$$$$$$

         N         M        IN       NSAM      NENS       NR
         4         1         2        60          1         4
 ND
 4
  IPRT  IPLT
    1
             PHI
1.0       1.0
          1.0
                    1.0       1.0
                              1.0
                  H
1.0
   R
0.0001
     COVW
.0001
           .0001
   GAMMA
0.5
1.0
          0.5
          1.0
          PKKM1
.0001                .0001
          .0001
.0001                .0001
                               .0001
          PRR(1)
.0001                .0001
          .0001
.0001                .0001
                               .0001
          PRR(2)
.0001                .0001
          .0001
.0001                .0001
                               .0001
          PRR(3)
```

```
.0001                 .0001
          .0001
.0001                 .0001
                                .0001
SIGV
.01
                         XHATZ - JET
                    0.3333                10.
                         XHATZ - INFANTRY1
                                                                0.00166667
                         XHATZ - INFANTRY2
10.                                       10.                   -0.00166667
                         XHATZ - HELICOPTER
12.                 -.05555555
                           XS - JET
                    0.3333                10.
                           XS - INFANTRY1
                                                                0.00166667
                           XS - INFANTRY2
10.                                       10.                   -0.00166667
                           XS - HELICOPTER
12.                 -.05555555
```

LIST OF REFERENCES

1. Dittmar, C. A. , Jr., The Application of Extended Kalman Filtering to the Position Locating Reporting System (PLRS), MSEE Thesis, Naval Postgraduate School, Monterey, California, 1975.

2. Sorenson, H. W., "Kalman Filtering Techniques," Advanced Control System, Vol. 3,Chapter 5, 1966.

INITIAL DISTRIBUTION

		No. Copies
1.	Chairman, Code 62 Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
2.	Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3.	Dr. H. A. Titus, Code 62Ts Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	2
4.	Capt. Bernard de Mahy, Jr. 1st MAW FPO San Francisco, California 96602	2
5.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2

168895
Thesis
D29845 de Mahy
c.1
Extended Kalman filtering applied to the position locating and reporting system (PLRS).

5 MAY 80 25783
8 NOV 83 29170

168895
Thesis
D29845 de Mahy
c.1
Extended Kalman filtering applied to the position locating and reporting system (PLRS).

thesD29845
Extended Kalman filtering applied to the
3 2768 001 02535 6
DUDLEY KNOX LIBRARY